

UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR



TRABAJO FIN DE GRADO

GENERADOR DE HOJAS DE CÁLCULO CORRECTAS

Cristina García Rufes

ENERO 2014-01-13

Proyecto: Generador de Hojas de Cálculo Correctas

No está permitida la reproducción total o parcial de este proyecto, ni su tratamiento informático, ni la transmisión de ninguna forma o por cualquier medio, ya sea electrónico, mecánico, por fotocopia, por registro u otros métodos, sin el permiso previo y por escrito del autor del mismo.

Si se quiere ascender por cuestas
empinadas,
es necesario, al principio, andar
despacio

(William Shakespeare)

GENERADOR DE HOJAS DE CÁLCULO CORRECTAS

Autor: *Cristina García Rufes*

Tutor: *Juan de Lara Jaramillo*

Miembros del Tribunal:

--

Fecha de lectura:

--

Calificación:

--

RESUMEN

Título: Generador de Hojas de Cálculo Correctas

Autor: Cristina García Rufes

Tutor: Juan de Lara Jaramillo

El objetivo de este proyecto es la implementación de una herramienta que genere hojas de cálculo Excel de forma correcta, a partir de un diagrama de clases, proponiendo así una solución a la amplia problemática del tratamiento de datos en las hojas de cálculo. Esta aplicación se adaptará a los diferentes usuarios que se estudiarán a lo largo del informe y hará uso de las funcionalidades más útiles de Excel.

Palabras Clave:

Hoja de cálculo, Excel, Generación correcta, Usuario, Usuario Final, Funcionalidad, Validación de datos, Diagrama de Clases, Características, Tipo de dato, Informes

ABSTRACT

The objective of this project is the implementation of a tool that generates Excel spreadsheets correctly, from a class diagram, thus proposing a solution to the wider problem of processing data in spreadsheets.

This application will adapt to the different users that will be explored throughout the report and will make use of the most useful features of Excel.

AGRADECIMIENTOS

A mis tíos, por haberme apoyado desde el principio y haberme animado cada día. Gracias por vuestro apoyo incondicional.

A mis abuelos, porque han sido una parte fundamental en mi vida y gracias a ellos he llegado a ser lo que soy hoy.

A mis compañeros de universidad, mención especial para Dagmara y Mario por ser los mejores compañeros de prácticas que he tenido nunca, y para Sara, Raquel, Esther y Verónica por todos los buenos momentos. A todos, gracias por las largas tardes y noches de prácticas, por los días interminables de laboratorio y biblioteca, por la ayuda recibida todos estos años y por haber estado ahí.

A Helena y Patricia, por saber entender mi situación y aguantar los malos ratos. Habéis estado al principio y sobretodo en este final tan duro, así que, simplemente, gracias.

A las tres de siempre, Vicki, Miriam y Rocío, por todos estos años, por el apoyo y por no irnos nunca.

A todo el equipo de Management Solutions, por haberme ayudado a aprender tantísimo.

A mi tutor, Juan de Lara, por haberme brindado la oportunidad de realizar este proyecto gracias al cual he podido adquirir más conocimientos. Gracias por tu ayuda.

A Cris, por apoyarme cada día, por estar en los buenos y malos momentos, por animarme a seguir y por motivarme. Gracias y mil gracias más por estar siempre ahí.

Y por último a mi madre, porque nada de esto hubiera sido posible sin tu ayuda y sin tu apoyo. Gracias por no dejar que me rindiera, por tu paciencia y por haber estado siempre conmigo. Sin ti hubiera sido imposible.

Gracias a todas las personas que de una forma u otra han pasado por mi vida a lo largo de estos años. Gracias por todas las experiencias vividas y por haber hecho que aprenda cada día un poquito más.

Gracias a todos, de corazón.

ÍNDICE DE CONTENIDOS

GLOSARIO	XII
1. INTRODUCCIÓN	13
1.1. DETERMINACIÓN DE LA SITUACIÓN ACTUAL	13
1.2. MOTIVACIÓN	13
1.3. ALCANCE Y OBJETIVOS	14
1.4. ESTRUCTURA DEL DOCUMENTO	14
2. ESTUDIO PREVIO - HOJAS DE CÁLCULO EN EXCEL	15
2.1. DEFINICIÓN Y ESTRUCTURA	15
2.1.1. Estructura	15
2.2. FUNCIONALIDADES	17
2.2.1. Fórmulas y referencias	17
2.2.2. Representaciones y tipos de informes	17
2.2.3. Herramientas y aplicaciones	21
2.2.4. Funciones	22
2.2.5. Tratamiento de los datos	23
2.2.6. Importación de datos	24
2.2.7. Macros en VBA	24
2.3. TIPOS DE USUARIOS	26
2.4. PROBLEMÁTICA DE LAS HOJAS DE CÁLCULO	27
3. ESTADO DEL ARTE	29
4. INTRODUCCIÓN A LA SOLUCIÓN PROPUESTA	33
4.1. ANÁLISIS DEL PROBLEMA INICIAL	33
5. ANÁLISIS Y DISEÑO	38
5.1. ANÁLISIS	38
5.1.1. Especificación de requisitos	38
5.1.2. Casos de uso	48
5.1.2.1. Caso de uso: carga de la hoja de plantilla mediante aplicación	49
5.1.2.2. Caso de uso: carga de la hoja de plantilla mediante hoja de cálculo en Excel / Generación de hoja de cálculo	50
5.2. DISEÑO	52
5.2.1. Esquema general	52
5.2.2. Diseño de la aplicación	53
5.2.2.1. Arquitectura de tres capas	53
5.2.2.1.1. Ventajas de la arquitectura de tres capas	55

5.2.3.	<i>Diagrama de clases</i>	55
5.2.3.1.	Capa_BD	56
5.2.3.2.	Capa_Objeto_BD	56
5.2.3.3.	Capa Interfaz	56
5.2.4.	<i>Esquema de la base de datos</i>	58
6.	IMPLEMENTACIÓN	61
6.1.	PLATAFORMA .NET FRENTE A OTRAS ALTERNATIVAS	61
6.2.	BASE DE DATOS ACCESS FRENTE A OTRAS ALTERNATIVAS	62
6.3.	IMPLEMENTACIÓN	63
6.3.1.	<i>Clase Utils BD</i>	64
6.3.2.	<i>Clase ManejadorExcel</i>	64
6.3.3.	<i>Formularios – Interfaz gráfica</i>	65
7.	PRUEBAS Y CASOS DE ESTUDIO	67
7.1.	PROBLEMA INICIAL	67
7.2.	GESTOR DE VENTAS	85
7.3.	ANALIZADOR FINANCIERO	90
8.	CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO	94
8.1.	CONCLUSIÓN DEL TRABAJO REALIZADO	94
8.2.	CONCLUSIÓN PERSONAL	95
8.3.	POSIBLES MEJORAS	95
8.4.	LÍNEAS FUTURAS DE TRABAJO	96
9.	REFERENCIAS	97
ANEXOS		98
ANEXO 1 – IMÁGENES DIAGRAMA DE CLASES		98
<i>Clases de la Capa_BD</i>		98
<i>Clases Capa_Objeto_BD</i>		98
<i>Capa_Interfaz</i>		101

ÍNDICE DE FIGURAS

ILUSTRACIÓN 1 - HOJA DE CÁLCULO EXCEL.....	16
ILUSTRACIÓN 2 - PROYECTO VBA	16
ILUSTRACIÓN 3 - EJEMPLO TABLA ESTÁTICA.....	17
ILUSTRACIÓN 4 - EJEMPLO DE TABLA DINÁMICA 1	18
ILUSTRACIÓN 5 - EJEMPLO CONFIGURACIÓN TABLA DINÁMICA 1	18
ILUSTRACIÓN 6 - EJEMPLO TABLA DINÁMICA 2	19
ILUSTRACIÓN 7 - EJEMPLO CONFIGURACIÓN TABLA DINÁMICA 2	19
ILUSTRACIÓN 8 - EJEMPLO GRÁFICO ESTÁTICO	20
ILUSTRACIÓN 9 - EJEMPLO GRÁFICO DINÁMICO	20
ILUSTRACIÓN 10 - EJEMPLO APLICACIÓN DE OFICINA.....	21
ILUSTRACIÓN 11 - EJEMPLO DE FORMULARIO	22
ILUSTRACIÓN 12 - USO DE FUNCIONES EXCEL.....	23
ILUSTRACIÓN 13 - FILTRO DE DATOS.....	23
ILUSTRACIÓN 14 - VALIDACIÓN DE DATOS	24
ILUSTRACIÓN 15 - IMPORTACIÓN DE DATOS EN EXCEL	24
ILUSTRACIÓN 16 - MACRO EN VBA	25
ILUSTRACIÓN 17 - EJEMPLO DE PROYECTO VBA.....	26
ILUSTRACIÓN 18 - SOLUCIÓN DE GREGOR ENGELS Y MARTIN ERWING	29
ILUSTRACIÓN 19 - TABLA DE DATOS VITSL	30
ILUSTRACIÓN 20 – SINTAXIS DEL LENGUAJE	30
ILUSTRACIÓN 21 – ANALIZADOR DE CLASES	30
ILUSTRACIÓN 22 - REGLAS DE INFERENCIA	31
ILUSTRACIÓN 23 - GENERACIÓN DE CLASES DESDE HOJA DE CÁLCULO	31
ILUSTRACIÓN 24 - TABLA GENERADA DE UNA CLASE.....	31
ILUSTRACIÓN 25-PROBLEMA INICIAL.....	34
ILUSTRACIÓN 26- ATRIBUTOS EN EXCEL.....	34
ILUSTRACIÓN 27 - TIPOS DE DATOS EN EXCEL.....	35
ILUSTRACIÓN 28 - VALIDACIÓN DE DATOS DE TIPO LISTADO EN EXCEL	35
ILUSTRACIÓN 29 - DIAGRAMA DE CASOS DE USO	48
ILUSTRACIÓN 30 - ESQUEMA GENERAL DEL DISEÑO.....	52
ILUSTRACIÓN 31 - ESQUEMA BÁSICO DISEÑO TRES CAPAS	53
ILUSTRACIÓN 32 - ESQUEMA INTERACCIÓN ENTRE CAPAS	54
ILUSTRACIÓN 33- ESQUEMA RELACIONAL BD	60
ILUSTRACIÓN 34 - SOLUTION EXPLORER APLICACIÓN.....	63
ILUSTRACIÓN 35 - RESULTADO FINAL MACRO.....	64
ILUSTRACIÓN 36 – DESIGNER.....	65
ILUSTRACIÓN 37 - CODE	66
ILUSTRACIÓN 38 - PANTALLA DE INICIO.....	68
ILUSTRACIÓN 39 - PANTALLA GESTIÓN TIPOS DE DATOS	68

ILUSTRACIÓN 40- ALTA TIPO DE DATO DNI 1	69
ILUSTRACIÓN 41 - ALTA TIPO DE DATO DNI 2.....	69
ILUSTRACIÓN 42 - ALTA TIPO DE DATO CALIFICACIÓN 1	70
ILUSTRACIÓN 43 - ALTA TIPO DE DATO CALIFICACIÓN 2.....	70
ILUSTRACIÓN 44 - ALTA TIPO DATO NOTA 1	71
ILUSTRACIÓN 45 - ALTA TIPO DATO NOTA 2	71
ILUSTRACIÓN 46 - PANTALLA GESTIÓN DE PROYECTOS	72
ILUSTRACIÓN 47 - PANTALLA NUEVO PROYECTO	72
ILUSTRACIÓN 48 - ALTA CARACTERÍSTICAS ALUMNO	73
ILUSTRACIÓN 49 - CONFIGURACIÓN ALUMNO FINAL	73
ILUSTRACIÓN 50 - CONFIGURACIÓN HOJA PLANTILLA ALUMNO	74
ILUSTRACIÓN 51 - CONFIGURACIÓN CARACTERÍSTICAS HOJA PLANTILLA ALUMNO	74
ILUSTRACIÓN 52 - CARGA DE DATOS ALUMNO	75
ILUSTRACIÓN 53 - PRUEBA DATO INCORRECTO 1	75
ILUSTRACIÓN 54 - PRUEBA DATO INCORRECTO 2	76
ILUSTRACIÓN 55 - PRUEBA DATO INCORRECTO 3	76
ILUSTRACIÓN 56 - PRUEBA DATO INCORRECTO 4	77
ILUSTRACIÓN 57 - INSERCIÓN DATO CORRECTO	77
ILUSTRACIÓN 58 - HOJA DE CÁLCULO GENERADA	78
ILUSTRACIÓN 59 - INSERCIÓN DATO INCORRECTO HOJA DE CÁLCULO	78
ILUSTRACIÓN 60 - INSERCIÓN DATO DUPLICADO HOJA DE CÁLCULO	79
ILUSTRACIÓN 61 - INSERCIÓN ERRÓNEA DATO OBLIGATORIO HOJA DE CÁLCULO	79
ILUSTRACIÓN 62 - COPIA DE SEGURIDAD ALMACENADA	80
ILUSTRACIÓN 63 - FÓRMULA INCORRECTA	80
ILUSTRACIÓN 64 - AÑADIR OBJETO A OTRA CLASE.....	81
ILUSTRACIÓN 65 - RESULTADO FINAL CLASE EXAMEN	81
ILUSTRACIÓN 66- ERROR AL AÑADIR DOS REFERENCIAS AL MISMO OBJETO.....	82
ILUSTRACIÓN 67 - INSERCIÓN REFERENCIA INEXISTENTE.....	82
ILUSTRACIÓN 68 – AUTOCOMPLETAR	83
ILUSTRACIÓN 69 - INSERCIÓN NOTA INVÁLIDA.....	83
ILUSTRACIÓN 70 - FÓRMULAS Y DATOS ENUM	83
ILUSTRACIÓN 71 - VALIDACIÓN DE DATOS EN LISTA EXCEL	84
ILUSTRACIÓN 72 - FÓRMULAS AUTOMÁTICAS	84
ILUSTRACIÓN 73 - REFERENCIAS A OBJETO DE OTRA TABLA	84
ILUSTRACIÓN 74 – CONFIGURACIÓN CLASE PAÍS.....	86
ILUSTRACIÓN 75 - TABLA PAÍS APLICACIÓN	86
ILUSTRACIÓN 76 - TABLA PAÍS EXCEL	86
ILUSTRACIÓN 77 - CLASE INFORME DE VENTAS.....	87
ILUSTRACIÓN 78 - TIPO DE DATO DINERO VENTAS	88
ILUSTRACIÓN 79 - TIPO DATO PRODUCTO	88

ILUSTRACIÓN 80 - CARGA DEL INFORME DE VENTAS.....	89
ILUSTRACIÓN 81 - INFORME DINÁMICO VENTAS.....	89
ILUSTRACIÓN 82 - INFORME FINAL PAÍS.....	90
ILUSTRACIÓN 83 - CLASE AÑOS	91
ILUSTRACIÓN 84 - CLASE RATIO FINANCIERO.....	91
ILUSTRACIÓN 85 - CLASE MESES	91
ILUSTRACIÓN 86 - CLASE INFORMEFINAL	92
ILUSTRACIÓN 87 - EJEMPLO CARGA INFORMEFINAL	92
ILUSTRACIÓN 88 - EJEMPLO TABLA DINÁMICA INFORMEFINAL.....	93
ILUSTRACIÓN 89 - GRÁFICO DINÁMICO INFORMEFINAL	93
ILUSTRACIÓN 90 - CLASES CAPA BD	98
ILUSTRACIÓN 91 - CLASES CAPA_OBJETO_BD1.....	98
ILUSTRACIÓN 92 - CLASES CAPA_OBJETO_BD 2.....	99
ILUSTRACIÓN 93- CLASES CAPA_OBJETO_BD 3.....	100
ILUSTRACIÓN 94 – CLASES CAPA_INTERFAZ 1	101
ILUSTRACIÓN 95 - CLASES CAPA INTERFAZ 2	102
ILUSTRACIÓN 96 - CLASES CAPA INTERFAZ 3	103
ILUSTRACIÓN 97 - CAPA INTERFAZ 4	104
ILUSTRACIÓN 98 - CLASES CAPA INTERFAZ 5	105
ILUSTRACIÓN 99 - CLASES CAPA_INTERFAZ 6	106
ILUSTRACIÓN 100 - CAPA_INTERFAZ 7	107

ÍNDICE DE TABLAS

TABLA 1 - REQUISITO RSF_01	38
TABLA 2 - REQUISITO RSF_02	39
TABLA 3 - REQUISITO RSF_03	39
TABLA 4 - REQUISITO RSF_04	40
TABLA 5 - REQUISITO RSF_05	40
TABLA 6 - REQUISITO RSF_06	41
TABLA 7 - REQUISITO RSF_07	41
TABLA 8 - REQUISITO RSF_08	42
TABLA 9 - REQUISITO RSF_09	42
TABLA 10 - REQUISITO RSF_10	43
TABLA 11 - REQUISITO RSF_11	43
TABLA 12 - REQUISITO RSF_12	44
TABLA 13 - REQUISITO RSF_13	45
TABLA 14 - REQUISITO RSF_14	45
TABLA 15 - REQUISITO RSNF_01	46
TABLA 16 - REQUISITO RSNF_02	46
TABLA 17 - REQUISITO RSFN_03	47
TABLA 18- BD - ENTIDAD "PROYECTO"	58
TABLA 19- BD- ENTIDAD "OBJETO"	58
TABLA 20 - BD - ENTIDAD "CARACTERÍSTICA"	58
TABLA 21 – BD - ENTIDAD "TIPODATO"	58
TABLA 22 - BD - ENTIDAD "TIPODATO"	59
TABLA 23 - BD - ENTIDAD "TIPOPLANTILLA"	59

GLOSARIO

- **Hoja de cálculo:** tipo de documento que permite manipular datos numéricos y alfanuméricos dispuestos en forma de tablas compuestas por celdas.
- **Excel:** aplicación distribuida por Microsoft Office para hojas de cálculo.
- **Generación correcta:** hoja de cálculo generada con un sistema de validación de datos que permite la inserción de datos de forma correcta acorde a unas propiedades que han sido previamente configuradas.
- **Usuario:** persona que utiliza algún tipo de objeto. En este documento se aplica a la herramienta software de la que se esté hablando en el contexto.
- **Usuario final:** del inglés (“End User”) tipo de usuario que no es programador pero utiliza herramientas de desarrollo para crear o modificar objetos software.
- **Funcionalidad:** conjunto de características que hacen que algo sea práctico y utilitario. En este caso se aplica a aquellas características que hacen que una aplicación software sea útil.
- **Validación de datos:** verificar y controlar cada uno de los datos que se introducen en función de unas propiedades.
- **Clase:** es un tipo de concepto informático, de la programación orientada a objetos, que se utiliza como modelo para representar un sustantivo, como una persona, lugar o cosa.
- **Diagrama de clases:** es un tipo de esquema que describe la estructura de un sistema mediante sus clases.
- **Características:** a lo largo de este documento, en determinadas situaciones, hacen referencia a las propiedades de una clase.
- **Tipo de dato:** atributo que impone restricciones en los datos, en función de los valores que puede tomar y operaciones que puede realizar.
- **Informes:** diferentes representaciones de los datos en una hoja de cálculo.

1. Introducción

1.1. Determinación de la Situación actual

A partir de la segunda mitad del siglo XX, la informática se ha desarrollado muy rápidamente, convirtiéndose así en una parte esencial, tanto del ámbito social como del laboral. Actualmente existe una cantidad muy diversa de aplicaciones capaces de resolver problemas de todo tipo, desde lo más simple hasta lo más complejo. Se habla de pequeños programas para uso cotidiano hasta bases de datos de gran complejidad. El procesamiento de la información y el tratamiento de datos en formato digital, hacen de la informática una de las armas más potentes actualmente. Es por ello que su uso indebido o erróneo puede dar lugar a problemas de gran importancia.

Una de las herramientas que más se utiliza a nivel mundial es la hoja de cálculo. Las hojas de cálculo son una importante herramienta utilizada como aplicación de negocio de gran versatilidad para el tratamiento y análisis de datos. Son fundamentales a la hora de presentar resultados y conclusiones ya que permiten ahorrar mucho tiempo de elaboración y resolución de cálculos. Pueden utilizarse simplemente para almacenar datos o como fuentes de información, por lo que es esencial que los datos que se almacenen y se traten no den lugar a errores.

Las hojas de cálculo poseen la propiedad de ser como lenguajes de programación debido a la funcionalidad tan amplia que ofrecen con respecto a la inserción de fórmulas y métodos de cálculo, pero muchos usuarios no son programadores profesionales. Los usuarios finales, acaban convirtiéndose en programadores y hacen que la hoja de cálculo sea más propensa a errores. Otro dato es que, según (1), el 57% de los usuarios de Excel no ha recibido formación sobre las hojas de cálculo, lo que dificulta aún más la situación.

1.2. Motivación

Desde su momento de creación, las hojas de cálculo se volvieron indispensables en lo que al tratamiento de datos se refiere, lo que las vuelve igual de ventajosas que problemáticas. Tanto entidades financieras como pequeños negocios dependen del manejo de datos que se realice en ellas y por ello, es de vital importancia la corrección de los datos en las mismas.

Las hojas de cálculo son utilizadas por un amplio perfil de usuarios. El extenso abanico comienza en expertos matemáticos capaces de realizar modelos de datos financieros hasta usuarios comunes que simplemente quieren llevar las cuentas de su casa. Cada uno de estos usuarios está expuesto a los mismos riesgos ya que no es necesario tener un nivel alto de informática para utilizar una hoja de cálculo, pero tampoco lo evita un conocimiento experto.

La idea de este proyecto surge precisamente del hecho común de cometer errores cuando se está utilizando una hoja de cálculo. Estos errores pueden cometerse, por ejemplo, en la valoración de un activo, en la determinación de cuentas financieras, en el cálculo de las dosis de medicamentos o incluso en el tamaño de una viga de soporte de carga en ingeniería estructural. Según (1), dado que el riesgo de una hoja de cálculo está vinculado a las acciones de los individuos, se define como una subcategoría de “riesgo operacional”. Este artículo, explica además, que el 94% de las hojas de cálculo contienen errores.

La experiencia laboral en una consultora durante un año y medio ha permitido que el análisis a la problemática pueda ser más profundo debido al conocimiento y experiencia en hojas de cálculo bancarias y ha motivado la idea de encontrar una solución para aquellos clientes que dependen tanto de los datos que se almacenen en ellas.

Para poder proponer una solución, primeramente es necesario analizar las hojas de cálculo y entender su funcionamiento, con el objetivo de comprender su impacto en la sociedad. En concreto se analizarán las de Excel debido a que es la herramienta más utilizada a nivel mundial. Este análisis se centra en sus diferentes funcionalidades, problemática y tipos de usuario, desde expertos programadores hasta usuarios básicos. De forma adicional, también es útil ver qué tipos de herramientas existen en el mercado que ayudan a la mejora de la seguridad de las hojas de cálculo.

Este documento recoge toda la información que ha sido necesaria para idear la herramienta implementada que ha sido implementada y que da solución a la problemática estudiada.

1.3. Alcance y Objetivos

El objetivo de este proyecto es implementar una aplicación que permita la creación y manipulación de hojas de Excel de forma segura.

La herramienta desarrollada permitirá, a través de un diseño conceptual de modelo de datos, el diseño y producción de una hoja de cálculo correcta. Esta hoja de cálculo estará controlada por una serie de macros que se generarán de manera automática y de forma invisible para el usuario. Como resultado, la hoja creada asegurará que los datos que se introducen y se almacenan son válidos y que las referencias de unos datos a otros son correctas.

A parte del requisito principal de la seguridad, la herramienta permitirá una configuración previa de los datos y hará uso de algunas de las funcionalidades más útiles de las hojas de cálculo.

1.4. Estructura del documento

A continuación, el documento tratará los siguientes puntos:

- 2) Estudio Previo – Hojas de cálculo en Excel: apartado donde se analizarán las hojas de cálculo, sus funcionalidades, sus tipos de usuario y su problemática.
- 3) Estado del arte: se estudiarán las tecnologías actuales y las soluciones propuestas para tratar la problemática de las hojas de cálculo.
- 4) Introducción a la solución propuesta: se describirá la propuesta realizada para la implementación de la herramienta.
- 5) Análisis y diseño: Se describen los requisitos de la aplicación el esquema general que sigue, el diagrama de clases y el esquema de la base de datos.
- 6) Implementación: Se explica cómo se ha desarrollado la aplicación realizando una comparativa de las tecnologías utilizadas.
- 7) Pruebas: se exponen tres casos de prueba con diferentes tipos de usuario.
- 8) Conclusiones y futuras líneas de trabajo: punto que trata las conclusiones del trabajo realizado, las conclusiones personales y las propuestas de futuro para la aplicación.

2. Estudio Previo - Hojas de cálculo en Excel

Microsoft Office es actualmente una de las herramientas laborales más utilizadas, tanto en el ámbito de procesadores de texto como en hojas de cálculo. La razón de esto es el potente software que ofrece, dando lugar a aplicaciones de gran flexibilidad y con muchas posibilidades de uso.

Excel forma parte del paquete de Microsoft Office y es el software más utilizado en cuanto a hoja de cálculo. Su primera versión se publicó en 1985 para Mac y en 1987 para Windows. Algunas de las aplicaciones previas fueron Lotus 1-2-3 y Quattro Pro, pero debido a su lentitud Excel se consolidó rápidamente. Algunos competidores actuales de esta aplicación son Calc (de OpenOffice), Gnumeric (de Gnome Office) y Numbers (de Apple), sin embargo el uso de Excel está muy por encima de los anteriores. El motivo de que Excel se utilice por un gran número de personas es la facilidad de su uso y la gran cantidad de funcionalidades que ofrece. Según (2), es una herramienta líder en el mercado debido a su potente software y flexibilidad.

Para poder estudiar el impacto que han tenido las hojas de cálculo y la importancia de su uso es necesario primeramente analizar su estructura y sus principales funcionalidades.

Este punto comienza explicando su arquitectura y sus principales usos para finalizar concluyendo con la problemática derivada de su abanico de posibilidades.

2.1. Definición y estructura

Se puede entender una hoja de cálculo como un tipo de documento que permite manipular, almacenar y organizar la información a través de una tabla de datos. La estructura principal que se utiliza es un área de trabajo en forma matricial compuesta por un número determinado de filas y columnas. La unidad básica de información es lo que se conoce como “celda” y es el lugar donde se insertan los datos, siendo éstos constantes alfanuméricas o fórmulas dinámicas que se calculan en tiempo real.

2.1.1. Estructura

La aplicación se compone de los siguientes elementos:

- **Hoja:**
 - Unidad fundamental que define el área de trabajo, formada por la matriz de datos.
- **Libro:**
 - Unidad que contiene varias hojas relacionadas entre sí.

- **Datos:**
 - Información que contienen las celdas. Esta información puede estar en forma de:
 - Datos alfanuméricos: textos, números...
 - Funciones predefinidas para cálculos: normalmente son fórmulas de Excel.
- Otros elementos: propios de la interfaz gráfica de la herramienta, como por ejemplo:
 - Cuadro de nombres
 - Barra de títulos
 - Barra de herramientas
 - Barra de fórmulas

En la siguiente imagen se puede ver un ejemplo de una hoja de cálculo:

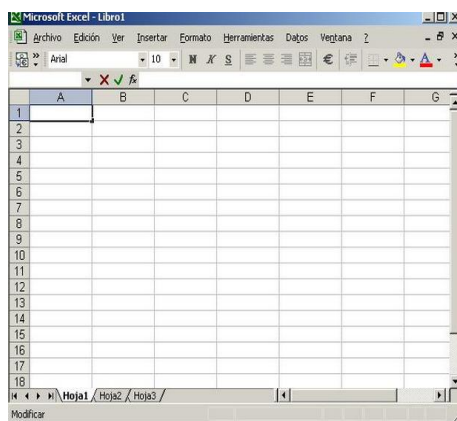


Ilustración 1 - Hoja de cálculo Excel

Todos los elementos anteriores forman parte de lo que se conoce como “Proyecto VBA”(Proyecto en Visual Basic), que básicamente es la estructura final que define la aplicación, compuesta por los objetos explicados anteriormente. Se explicará más detalladamente en el punto 2.2.7Macros en VBA.

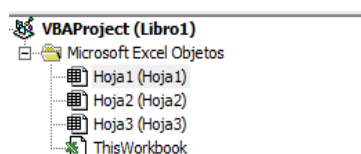


Ilustración 2 - Proyecto VBA

2.2. Funcionalidades

Las hojas de cálculo ofrecen una amplia gama de posibilidades. Actualmente y en su forma más tradicional, es muy común su uso para manejo de grandes volúmenes de datos, como si fuesen una base de datos, operaciones complejas y presentaciones de distintos tipos de informes. Gracias a su versatilidad es muy sencillo realizar cualquier tipo de modelo ya sea para el diseño y cálculo de estructuras civiles, contabilidad de una empresa, diseños de modelos matemáticos, planificación de presupuestos, etc.

A continuación se detallarán algunas de sus funcionalidades más importantes.

2.2.1. Fórmulas y referencias

La prestación fundamental es que **las fórmulas tienen la capacidad de recalcularse**, es decir, que se evalúan de forma dinámica. A medida que las celdas o los datos a los que hacen **referencia** cambian, se actualizan a **tiempo real**. Para el tratamiento de grandes cantidades de datos y generación de resultados es algo esencial ya que ahorra tiempo y costes.

2.2.2. Representaciones y tipos de informes

Otra de las funcionalidades principales es la posibilidad de generar diferentes representaciones de los datos a través de tablas y gráficos, que es posible diseñar y editar con la interfaz que se desee. Algunas de estas **representaciones** son:

- **Tablas estáticas:**
 - o Formadas por N filas y M columnas.

	A	B	C	D	E	F
1	Ventas Totales					
2	Nombre País	Producto	Fecha Incorporación	PrecioVenta	CantidadVendida	Total
3	Canadá	Vino	01/07/2014 0:00	100	5	500,00
4	México	Vino	01/06/2014 0:00	80	4	320,00
5	EEUU	Vino	21/10/2013 0:00:00	200	10	2.000.000
6	Francia	Vino	22/10/2013 0:00:00	150	6	900,00
7	Alemania	Vino	16/09/2013 0:00:00	80	4,5	360,00
8	Canadá	Soda	22/10/2013 0:00:00	200	100	20.000.000
9	México	Soda	11/12/2013 0:00	50	30	1.500.000
10	EEUU	Soda	01/12/2014 21:16	120	50	6.000.000
11	Canadá	Agua	01/06/2014 0:00	350	600	210.000.000
12	EEUU	Agua	01/06/2014 0:00	600	120	72.000.000
13						

Ilustración 3 - Ejemplo tabla estática

- **Tablas dinámicas:**
 - o Permiten mover y modificar las columnas y filas que forman el informe. Con ello, a partir de los mismos datos se pueden crear diferentes tipos de reportes.
 - Es una de las funcionalidades más útiles que presenta Excel ya que crea una variedad muy amplia de representación de datos. Permite además utilizar fórmulas diferentes de cálculo generando resultados por filas y por columnas.

Este tipo de tablas está provista de:

- Filtros
- Datos por fila
- Datos por columna
- Fórmula aplicada al total de filas (por ejemplo, suma por filas)
- Fórmula aplicada al total de columnas (por ejemplo, suma por columnas).

Estas tablas se usan a modo de visualización y no de inserción o modificación de datos, ya que toman como referencia una tabla estática.

Tomando de base el ejemplo de la Ilustración 3 – Ejemplo de tabla estática, se pueden crear diferentes tipos de tabla dinámica partiendo de este informe. Por ejemplo, se podría ver la cantidad vendida de los diferentes productos filtrado por país:

	A	B	C
1	Nombre País	(All)	
2			
3	Row Labels	Sum of CantidadVendida	
4	Agua	720	
5	Soda	180	
6	Vino	29,5	
7	(blank)		
8	Grand Total	929,5	
9			
10			

Ilustración 4 - Ejemplo de tabla dinámica 1

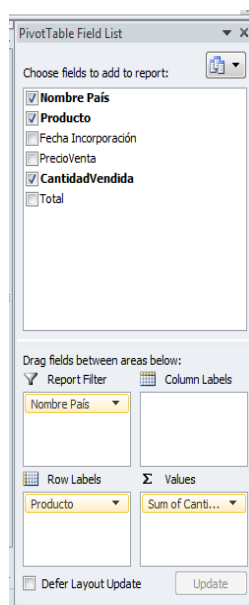


Ilustración 5 - Ejemplo configuración tabla dinámica 1

Otro de los tipos de tabla que se podría construir sería, por ejemplo, filtrando por fecha de incorporación y viendo qué productos se han vendido por país para esa fecha:

	A	B	C	D	E	F
1	Fecha Incorporación	(All)				
2						
3	Sum of CantidadVendida	Column Labels				
4	Row Labels	Agua	Soda	Vino	(blank)	Grand Total
5	Alemania			4,5		4,5
6	Canadá	600	100	5		705
7	EEUU	120	50	10		180
8	Francia			6		6
9	México		30	4		34
10	(blank)					
11	Grand Total	720	180	29,5		929,5
12						

Ilustración 6 - Ejemplo tabla dinámica 2

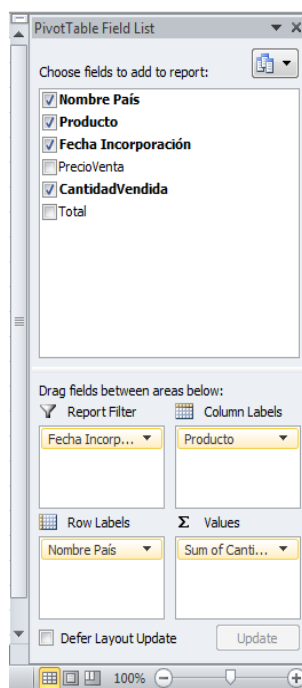


Ilustración 7 - Ejemplo configuración tabla dinámica 2

Como se puede observar se puede crear una variedad muy amplia de informes sin necesidad de programar ninguna fórmula, tan solo colocando las filas, columnas y filtros de la manera que se desee y partiendo de una única tabla de datos.

- **Gráficos:**

- Al igual que las tablas, pueden ser estáticos o dinámicos.
- Existe una gran cantidad de tipos, como por ejemplo de barras, de líneas, de tarta...
- Los gráficos se usan a modo de visualización de datos y se pueden generar muchos tipos de informes de datos.
- Partiendo de la Ilustración 3 - Ejemplo tabla estática, un gráfico estático que podría generarse es el siguiente:

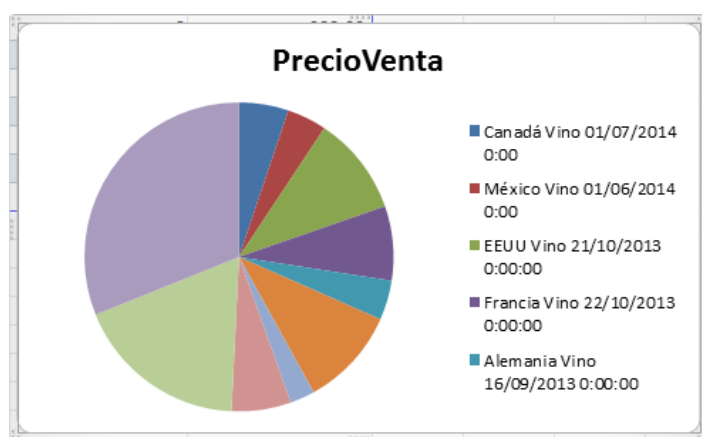


Ilustración 8 - Ejemplo gráfico estático

Como gráfico dinámico se podría crear un gráfico partiendo de la tabla dinámica Ilustración 6 - Ejemplo tabla dinámica 2:

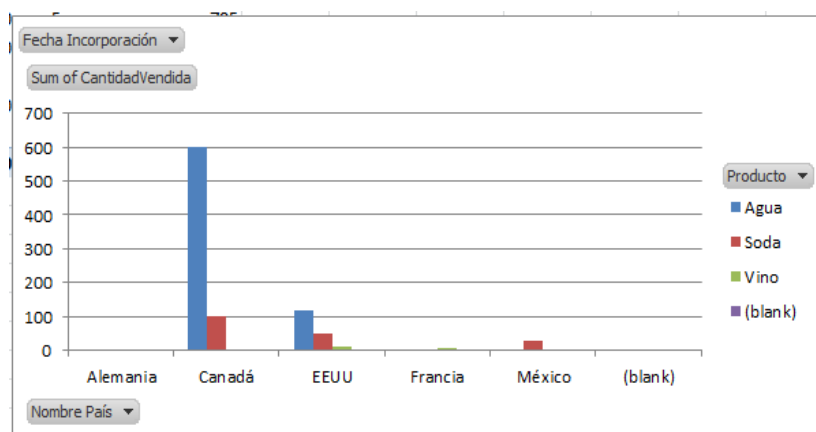


Ilustración 9 - Ejemplo gráfico dinámico

Como se puede observar, tomando como referencia el mismo origen de datos, se pueden crear una diversa variedad de tipo de informes, lo que abre un abanico de posibilidades a las hojas de cálculo.

2.2.3. Herramientas y aplicaciones

Excel ofrece también la posibilidad de crear sus propias aplicaciones, mediante:

- **Interfaces gráficas:**

- Creadas a partir de botones, campos de texto y una gran variedad de elementos gráficos que ofrece Excel. Gracias a la inclusión de estos elementos en las hojas, se pueden crear, por ejemplo, aplicaciones de oficina:

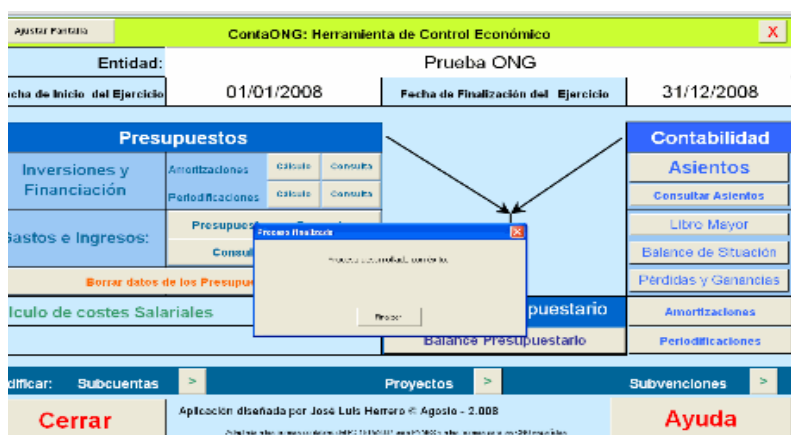


Ilustración 10 - Ejemplo aplicación de oficina

La aplicación que se muestra en el ejemplo corresponde a una hoja provista de botones que realizan acciones. No deben confundirse con formularios.

- **Formularios:**

- Un formulario es un tipo de aplicación que se diseña con un formato y una estructura estándar que facilita la interacción con el usuario. Los formularios se pueden crear a partir de código VBA mediante instrucciones y elementos gráficos.

Un ejemplo de formulario puede ser:

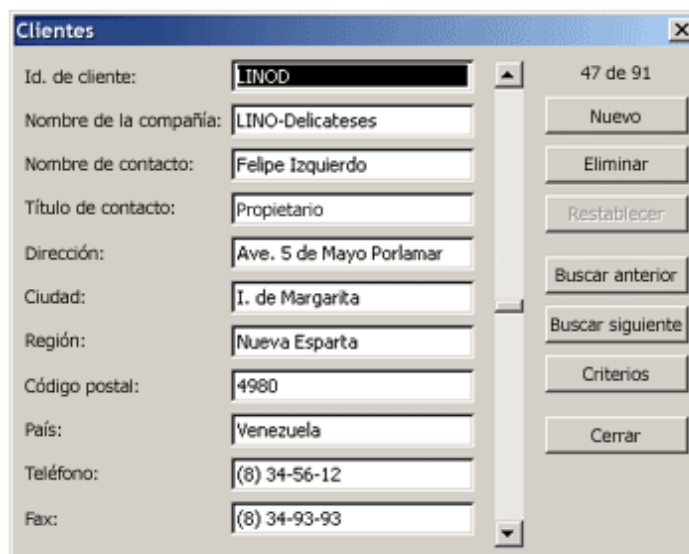


Ilustración 11 - Ejemplo de formulario

- Los formularios se usan normalmente para configurar propiedades de la hoja o de los datos de forma sencilla. Se usan también para realizar cargas y modificaciones de datos. Se puede entender como el punto intermedio entre el usuario y la hoja.

2.2.4. Funciones

Como herramienta de cálculo las hojas de Excel ofrecen una gran variedad de **funciones** que se pueden utilizar, desglosadas en los siguientes tipos:

- Funciones de complementos y automatización
- Funciones de cubo
- Funciones de base de datos
- Funciones de fecha y hora
- Funciones de ingeniería
- Funciones financieras
- Funciones de información
- Funciones lógicas
- Funciones de búsqueda y referencia
- Funciones matemáticas y trigonométricas
- Funciones estadísticas
- Funciones de texto

Un ejemplo muy utilizado de funciones en Excel, son las sumas condicionales, donde se suma si se cumple una cierta propiedad. En el caso del ejemplo siguiente, se sumarán las ventas para aquellas regiones que sean “Centro”.

	A	B	C	D	E	F
1	Nombre	Región	Ventas			
2	Juan	Norte	2.000 €		Total Ventas Centro	22.200 €
3	Pedro	Sur	3.500 €		Formula	=SUMAR.SI(B2:B10;"Centro";C2:C10)
4	Luis	Centro	1.200 €			
5	Carlos	Centro	4.000 €			
6	Manuel Tomás	Norte	5.000 €			
7	Hugo	Norte	6.000 €			
8	Sergio	Sur	7.000 €			
9	Bruno	Centro	8.000 €			
10	Diego	Centro	9.000 €			
11						

Ilustración 12 - Uso de funciones Excel

2.2.5. Tratamiento de los datos

Otras funciones útiles son el uso de **filtrado de datos** para tablas con numerosos datos. Permiten una visualización mejorada.

Marca	Modelo	Color	Cantidad
Opel	Corsa	(Todas)	1
Opel	Corsa	(Las 10 más...)	4
Opel	Astra	(Personalizar...)	3
Opel	Astra	Azul	2
Opel	Astra	Blanco	6
Opel	Astra	Gris	5
Opel	Vectra	Negro	3
Opel	Vectra	Rojo	4
Renault	Megane	Verde	2
Renault	Clio	Verde	6
Renault	Clio	Azul	7
Volvo	S40	Azul	2
Ford	Mondeo	Blanco	2
Ford	Mondeo	Negro	

Ilustración 13 - Filtro de datos

La **validación de datos** se usa para limitar y controlar los datos que el usuario inserta en una celda, permitiendo así definir restricciones propias para impedir que los usuarios escriban datos incorrectos.

	A	B	C	D	E	F
1			Entrada de presupuesto: marketing			
2			Cuenta	Real	Proyectado	
3			Costos de empleados			
4			110 Salario	45.328	60.000	
5			120 Impuestos/Hacienda/Estado	15.997	25.000	
6			140 Plan de pensiones	6.249	8.000	
7			160 Comisiones/Bonos	2.720	4.000	
8						
9						
10						
11						
12						
13						
14						
15						

Ilustración 14 - Validación de datos

2.2.6. Importación de datos

Es posible realizar **cargas de datos** mediante conexiones a bases de datos externas, ficheros de texto, etc, lo que amplía su rango de posibilidades.

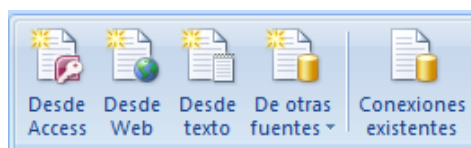


Ilustración 15 - Importación de datos en Excel

2.2.7. Macros en VBA

Visual Basic es un lenguaje de programación dirigido por eventos (es decir, manejados por los sucesos que ocurren en el sistema) con el objetivo de simplificar y facilitar la programación a través de elementos gráficos para la creación de interfaces de usuario. Provee facilidades para el desarrollo de aplicaciones de bases de datos y contiene un entorno de desarrollo, un depurador, un compilador y un editor de interfaces gráficas.

Visual Basic for Applications (VBA) es un subconjunto de Visual Basic y se utiliza para programar macros en aplicaciones Microsoft, permitiendo así ampliar la funcionalidad de estas herramientas.

Una macro en Excel es un tipo de código desarrollado en VBA que ejecuta instrucciones programadas en la hoja de cálculo, cuyo objetivo es automatizar determinadas operaciones. Gracias a esto, dentro de la misma aplicación de Excel, se pueden programar aplicaciones que se ejecutan por detrás.

Algunos usos comunes de macros son:

- Personalización de estilos y formatos
- Crear nuevas funciones para resolución de cálculos
- Accesos a bases de datos
- Formularios con interfaz gráfica

En la imagen siguiente se muestra un pequeño ejemplo de una macro que se ejecuta al dar click a un botón. Esta macro abre el archivo CPI.xls, selecciona la celda A1 y escribe en ella el valor “123”.

Como se puede observar, Excel tiene su propio editor de VBA donde escribir el código de las macros.

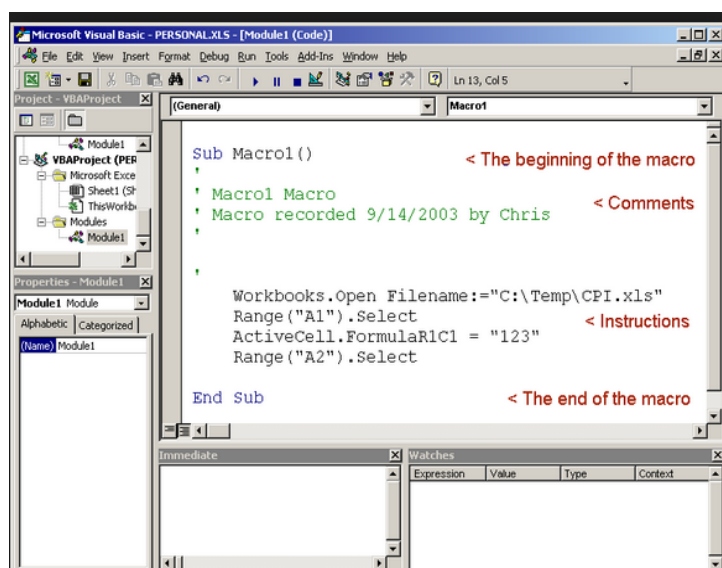


Ilustración 16 - Macro en VBA

En el punto 2.1.1 Estructura se habló del “Proyecto VBA” como conjunto de todos los elementos que forman parte de la hoja de cálculo. En este proyecto se pueden escribir macros en las propias hojas y además se permite crear “Módulos” (ficheros con código VBA que define el desarrollador), formularios con interfaz gráfica, etc.

A continuación se muestra un proyecto empresarial realizado en Visual Basic con Excel como ejemplo del amplio abanico de posibilidades que ofrecen las macros:

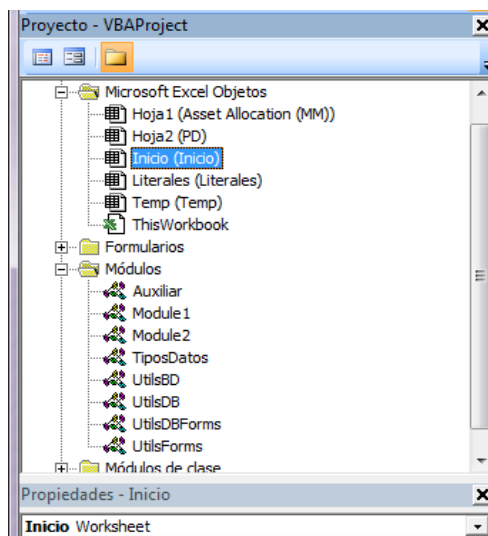


Ilustración 17 - Ejemplo de proyecto VBA

Este proyecto está dividido en 3 carpetas:

- **Objetos de Microsoft Excel:** objetos iniciales que forman parte de la estructura propia de una aplicación Excel: libro y hojas.
- **Formularios:** aplicaciones con interfaz gráfica que crea el propio programador.
- **Módulos:** ficheros de código donde se insertan las instrucciones de las macros.
 - o Estos ficheros se pueden administrar en diferentes subcarpetas y se pueden crear tantos módulos como necesite el programador.

2.3. Tipos de usuarios

En común con otras aplicaciones, se pueden distinguir los siguientes tipos de usuario para las hojas de cálculo de Excel:

- **Principiante:** aquel que usa las funciones más sencillas de Excel para algunos cálculos simples. Necesita de la interfaz y modelo de la hoja, pero no sabe utilizar la mayor parte de las funcionalidades.
- **Intermedio:** conoce más funciones que el usuario anterior y utiliza modelos de tablas, gráficos, etc. para generar informes. Usa determinados filtros y validaciones de datos.
- **Usuario avanzado:** crea modelos propios de cálculo, conoce y utiliza todas las funcionalidades de Excel (rangos dinámicos, macros, filtros avanzados, formatos condicionales...).

El libro (3), define 5 tipos de usuario Excel, describiéndolos de la siguiente forma:

- **Usuario básico y usuario avanzado (basic Excel user, Excel power user):** se corresponden a los usuarios descritos anteriormente.
- De forma muy interesante, separa a los programadores de VBA de este tipo inicial, creando las siguientes categorías:
 - o **Programador VBA (VBA Developer):** uso intensivo de código VBA sin conocer las soluciones alternativas a la programación.
 - o **Programador Excel (Excel Developer):** al contrario que el anterior, potencia las funcionalidades de Excel añadiendo código VBA de manera eficaz y sencilla. Desarrolla modelos complejos pero desconoce otro tipo de lenguajes.
 - o **Programador profesional:** capaz de interactuar con Excel, macros y otros lenguajes de programación.

De una forma casi inconsciente, el usuario de Excel puede convertirse en un programador de aplicaciones ya que desarrolla las capacidades suficientes para implementar código VBA, lo que le da ciertas ventajas sobre el resto. Al igual que tiene más ventajas, también tiene inconvenientes, ya que se convierte en un usuario peligroso debido a los conocimientos en manipulación de datos. Este tipo de usuarios se definen como “**Usuario final**”, de acuerdo a (3):

- (4) *“El desarrollo “Usuario final” (del inglés “End User Development” puede definirse como un conjunto de métodos, técnicas y herramientas que permite a los usuarios de sistemas de software, que no son desarrolladores, crear, modificar y ampliar artefactos de software.”*

Es decir, que personas que no son profesionales de esta rama y carecen de conocimientos de lenguajes de programación, tienen las capacidades para modificar e interactuar con objetos de software.

Debida a la característica de las hojas de cálculo de permitir a usuarios escribir sus propios programas, muchos de estos acaban convirtiéndose en “**desarrolladores**”.

2.4. Problemática de las hojas de cálculo

A pesar de todas las ventajas que se han visto anteriormente, existe un alto riesgo al usar hojas de cálculo. Son una herramienta muy potente pero a la vez está expuesta a muchos riesgos.

Según el artículo (5), Excel se puede considerar el programa más peligroso del mundo, debido a que *“se ha convertido en algo tan complejo y que usa de forma tan apresurada que en realidad ya nadie lo domina por completo”*.

En este artículo se hacen referencias al Comité de Supervisión Bancaria de Basilea (BCBS) y a la Autoridad de Servicios Financieros(FSA) del Reino Unido por la necesidad de medidas de control para la introducción de datos incorrectos, falsos e incluso fraudulentos.

James Kwak, en (6), explica el caso del Departamento de Inversión de un banco donde él trabajaba que necesitaba un programa nuevo para el cálculo del riesgo de sus activos más peligrosos. Encargaron el desarrollo a un matemático experto en cálculo y desarrollador de

modelos, quien ideó la operación a través de una serie de hojas de cálculo de Excel que debían de ser completadas manualmente, en un proceso de copia de una hoja a otra.

El uso incorrecto de esta aplicación, estalló en un escándalo debido a un agujero de 13 billones de dólares que el trader London Whale había causado en JP Morgan. Lo que causó este problemático error fue una fórmula incorrecta que se llevó de una hoja a otra.

Otro caso curioso es el nombrado en (7), donde Thomas Herndon, un estudiante de la universidad de Massachusetts, encontró grandes e importantes fallos en la hoja de cálculo que utilizaban las economistas Carmen Reinhart y Kenneth Rogoff en un artículo de revista que hablaba y justificaba algunos programas de austeridad europeos.

Esta situación lleva a pensar que los programadores de aplicaciones para Excel pueden desarrollar, desde sistemas automatizados capaces de autorregularse hasta aplicaciones que fallen de formas vertiginosas. Y ya no sólo los programadores, sino cualquier persona que cumpla una hoja de forma incorrecta en la que no se hagan validaciones de datos puede generar pérdidas millonarias.

Gregor Engels y Martin Erwig en (8) hablan de que el problema está en que los usuarios finales no llegan a comprender del todo la funcionalidad de las hojas de cálculo. A esta falta de conocimiento se le suma la libertad y la flexibilidad que ofrecen, lo que amplía la creación de hojas de cálculo erróneas. Critican la falta de un nivel de abstracción igual o parecido al de los lenguajes de programación actuales y el hecho de que las partes de código no estén separadas de los datos, es decir, que el mismo proyecto de código se encuentre en el mismo libro que los datos. Esto dificulta la reutilización de las hojas y amplía el abanico de errores.

Una posible solución podría ser dejar de utilizar hojas de cálculo. Analizando la situación, se puede pensar que es una aplicación que está expuesta a demasiados riesgos y por tanto no es válida. El problema es que la mayor parte de las cuentas bancarias y financieras de hoy en día dependen de la existencia de estas hojas de cálculo, ya que, si las éstas no existiesen, una gran parte de lo que hacen los mercados financieros no podría llevarse a cabo. No habría, por ejemplo, obligaciones de deuda colateralizada (CDO) ni seguros de impago de deuda (CDS).

La complejidad de la situación se da cuando, sin Excel ni hojas de cálculo no se podrían resolver todos los problemas financieros que se dan actualmente. Es un software realmente peligroso pero extremadamente necesario.

La propuesta que se hace en este proyecto trata de controlar y minimizar los riesgos que supone el uso de las hojas de cálculo para problemas tan complejos como los tratados. Parte de esta solución se basa en realizar previamente un diseño conceptual donde se indiquen los tipos de datos que van a insertarse en las hojas. De esta forma se disminuirá el riesgo de cometer este tipo de fallos.

3. Estado del Arte

Analizando los riesgos descritos en los puntos anteriores, la pregunta que se plantea es, ¿existen soluciones tecnológicas que ayuden a evitar este tipo de problemas? La respuesta es no. No existe una solución concreta ni una herramienta que se use a nivel mundial para evitar el uso indebido de las hojas de cálculo.

Según (9), desde hace no muchos años se están abriendo nuevas líneas de mercado que dan soluciones técnicas al mantenimiento de hojas de cálculo. *Ventana Research*, una empresa de asesoramiento líder, estimó que la inversión en éste área de mercado comenzó en 15 millones de dólares en 2006 y ascendió a 500 millones en 2011, lo que concluye que la seguridad en las hojas de cálculo se está volviendo cada vez más importante. Este documento, categoriza las soluciones técnicas en tres grupos:

- **Mantenimiento y control de hojas de cálculo:** actualización y mantenimiento de versiones, proporcionar métodos de control, realizar auditorías...
- **Búsqueda e investigación:** análisis automatizados de redes o servidores específicos para generar inventarios de todas las hojas de cálculo y recoger metadatos para facilitar la realización de una evaluación de riesgos.
- **Auditorías:** utilización de herramientas automatizadas para ayudar en la revisión de las auditorías.

En (10), se estudian métodos de ingeniería del software para evitar posibles errores. Uno de ellos es el uso de etiquetas para formalización de reglas y así evitar errores en la distinción de tipos básicos. La otra propuesta se basa en el reporte de errores a través de herramientas “add-in” implementadas para Excel, coloreando celdas inválidas que violen las reglas de la formalización de etiquetas.

Gregor Engels y Martin Erwig en (8) proponen una solución a la problemática nombrada en “2.4. Problemática de las hojas de cálculo”. Generan hojas de cálculo a partir de una descripción de alto nivel a través de una herramienta llamada “ViTSL”, siguiendo el siguiente proceso:

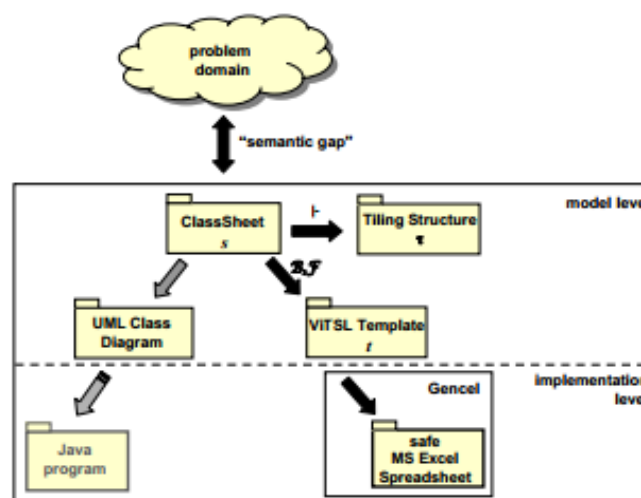


Ilustración 18 - Solución de Gregor Engels y Martin Erwing

Mediante la herramienta de “ViTSL” leen y traducen un diagrama de clases y crean una hoja de cálculo en “ViTSL” que generará después en una hoja de Excel sin errores. Como se verá en el punto “**¡Error! No se encuentra el origen de la referencia.¡Error! No se encuentra el origen de la referencia.**” una parte de la aplicación implementada se basa en esta idea.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	com_code	upc	description	size	case	nitem	store	week	move	qty	price	onsale	profit	ok
2	653	1111140009	DOVE DISH LIQUID	42 OZ	9	2851281	100	383	16	1	2.19		7.01	1
3	653	1111140009	DOVE DISH LIQUID	42 OZ	9	2851281	100	384	7	1	2.19		3.07	1
4	653	1111140009	DOVE DISH LIQUID	42 OZ	9	2851281	100	385	15	1	2.19		6.57	1
5														
6	654	1111165003	SUNLIGHT AUTO GEL	88 OZ	6	2857061	100	390	6	1	3.75		4.50	1
7	654	1111165003	SUNLIGHT AUTO GEL	88 OZ	6	2857061	100	391	11	1	3.39	S	7.46	1
8	654	1111165003	SUNLIGHT AUTO GEL	88 OZ	6	2857061	100	392	8	1	3.75		6.00	1
9														

Ilustración 19 - Tabla de datos ViTSL

Como se puede observar, la tabla que se ha implementado mediante esta herramienta tiene una interfaz muy parecida a la de las hojas de cálculo de Excel, lo que facilita al usuario la carga y manipulación de datos.

La herramienta implementada contiene un lenguaje lógico capaz de etiquetar y clasificar los elementos del diagrama de clases.

$f \in Fml$	$::= \varphi \mid n.a \mid \varphi(f, \dots, f)$	(formulas)
$b \in Block$	$::= \varphi \mid a = f \mid b \mid b \mid b^{\wedge} b$	(blocks)
$\ell \in Lab$	$::= h \mid v \mid .n$	(class labels)
$h \in Hor$	$::= n \mid \underline{n}$	(horizontal)
$v \in Ver$	$::= \overline{n} \mid \overline{\underline{n}}$	(vertical)
$c \in Class$	$::= \ell : b \mid \ell : b^{\downarrow} \mid c^{\wedge} c$	(classes)
$s \in Sheet$	$::= c \mid c^{\rightarrow} \mid s \mid s$	(sheets)

Ilustración 20 – Sintaxis del lenguaje

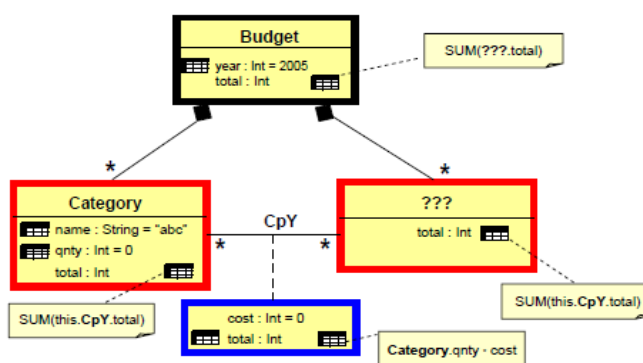


Ilustración 21 – Analizador de clases

Aplicando reglas de inferencia a los elementos que encuentran en el diagrama, generan una tabla resultante donde introducir los datos:

$$\begin{array}{c}
 \text{BASE} \frac{}{\vdash \ell : \beta :: \blacksquare} \quad \text{ROW} \frac{\vdash c :: \tau}{\vdash h : \beta | c | h : \beta' :: [\tau]} \\
 \text{ROW}^* \frac{\vdash c :: \tau}{\vdash h : \beta | c^- | h : \beta' :: [\tau]} \quad \text{COL} \frac{\vdash c :: \tau}{\vdash v : \beta \wedge c \wedge v : \beta' :: \langle \tau \rangle} \\
 \text{HOR} \frac{\vdash s_1 :: \tau_1 \quad \vdash s_2 :: \tau_2 \quad \uparrow \tau_1 = \uparrow \tau_2}{\vdash s_1 | s_2 :: \tau_1 | \tau_2} \\
 \text{VER} \frac{\vdash c_1 :: \tau_1 \quad \vdash c_2 :: \tau_2 \quad \overleftarrow{\tau_1} = \overleftarrow{\tau_2}}{\vdash c_1 \wedge c_2 :: \tau_1 \wedge \tau_2} \\
 \begin{array}{l}
 c'_1 = v : \beta_1 \wedge c_1 \wedge v : \beta_2 \\
 c'_2 = v' : \beta_3 \wedge c_2 \wedge v' : \beta_4 \\
 c'_3 = v : \beta_5 \wedge c_3 \wedge v : \beta_6 \\
 \vdash c'_1 :: \langle \tau \rangle \quad \vdash c'_2 :: \langle \tau \rangle \quad \vdash c'_3 :: \langle \tau \rangle \\
 \vdash c_1 | c_2 | c_3 :: [\tau]
 \end{array} \\
 \text{TAB} \frac{}{\vdash c'_1 | c'_2 | c'_3 :: [\tau]}
 \end{array}$$

Ilustración 22 - Reglas de inferencia

De forma contraria, en (11), se generan diagramas de clases desde una tabla de datos desde una hoja de cálculo a través del siguiente proceso:

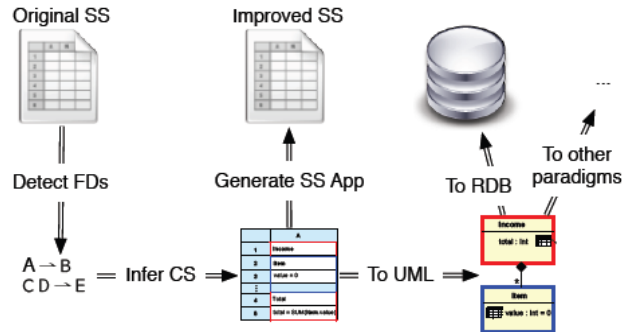


Ilustración 23 - Generación de clases desde hoja de cálculo

Aplicando reglas de inferencia de forma contraria al proceso anterior, la aplicación es capaz de deducir relaciones entre los datos e implementarlas en forma de tabla:

	A			...		
1	R			Mkey		
2				M ₁	...	M _r
3				m ₁ = M . M ₁	...	m _r = M . M _r
4	Nkey			R ₁	...	R _x
5	N ₁	...	N _t	r ₁ = d r ₁	...	r _x = d r _x
6	n ₁ = N . N ₁	...	n _t = N . N _t			
⋮						

Ilustración 24 - Tabla generada de una clase

La importancia de estas aplicaciones recae en el hecho de que relacionar hojas de cálculo con lenguajes de alto nivel, algo que se ha tomado como referencia para este proyecto, como se verá en apartados posteriores.

De otra forma diferente y más orientado a la tipología y conocimiento del cliente, algunas empresas de desarrollo y consultoras tienen expertos desarrolladores en VBA que se dedican a estudiar las necesidades del cliente y programar hojas para aplicaciones determinadas. Parte de estas aplicaciones están orientadas a banca y finanzas, para tipos de clientes muy concretos y muy acostumbrados al manejo de hojas de cálculo. Las propias hojas de Excel pasan a ser aplicaciones debido a la gran parte de código que incluyen siendo capaces de interactuar con bases de datos, realizar cálculos complejos y teniendo una interfaz de usuario provista de botones, menús, etc.

4. Introducción a la solución propuesta

Analizando la información anterior, se puede concluir que:

- Las hojas de cálculo son una herramienta fundamental para la resolución de problemas de muchos tipos, y existe una gran dependencia hacia ellas.
- Se adaptan a muchos tipos de usuario, a diferentes niveles de conocimiento informático.
- Cometer un error en la manipulación de los datos en una hoja de cálculo implica problemas de gran impacto.
- Son necesarias herramientas que ayuden a controlar los datos que se introducen y que funcionen como sistemas de seguridad.

4.1. Análisis del problema inicial

Para poder proporcionar una solución viable al problema de seguridad que se plantea, es necesario partir de un ejemplo concreto. El objeto de estudio será la idea inicial que dio lugar a este proyecto:

- *Es necesaria una hoja de Excel donde poder almacenar las calificaciones de los alumnos correspondientes a una asignatura, asegurando que no se inserten alumnos con un formato incorrecto y que las notas sean números del 1 al 10.*

Cualquier persona con un cierto nivel en **programación orientada a objetos** plantearía la solución de la siguiente forma:

- Diseño de clases:
 - Clase alumno, formada por:
 - DNI – char [8]
 - nombre- string
 - apellidos- string
 - Clase Examen, formada por
 - Alumnos - alumno [N]
 - nota 1 - entero
 - nota 2 – entero
 - nota final – Nota
 - Enumerado Nota:
 - Aprobado, Suspenso

El diagrama de clases correspondiente sería:

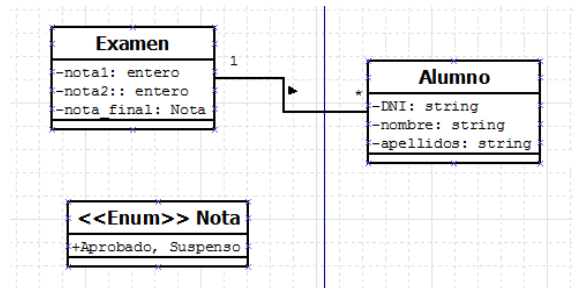


Ilustración 25-Problema Inicial

Si la resolución tiene que ser a partir de una hoja de Excel, podría representarse una clase como una hoja, lo que lleva a un **primer requisito**:

- **Posible diseño conceptual.**

- Clase → hoja

En la programación orientada a objetos, una clase está formada por atributos (características, propiedades). ¿Es aplicable a una hoja Excel? Si hubiera que cumplimentar una hoja de Excel con todos los alumnos de una asignatura, una posible forma sería esta:

DNI	Nombre	Apellidos	
70812937P	Cristina	García Rufes	
12345678X	Raquel	Martín Gómez	
91827490U	Susana	Fernández Rodríguez	

Ilustración 26- Atributos en Excel

Debido a la interfaz de trabajo proporcionada por Excel, una hoja de cálculo se rellena a través de filas y columnas, donde, se puede encontrar la correspondencia entre los atributos de una clase con las columnas de una hoja. Así pues, el **segundo requisito** será:

- **Configuración de los atributos de una clase**

- Atributos → Columnas de la hoja

En un diseño orientado a objetos, es necesario definir el tipo que van a tener los atributos de la clase. Excel también permite definir los datos que van a introducirse en las celdas, por lo que será necesario adecuar los tipos de dato de cada columna a los correspondientes en un diseño de diagrama de clases, lo que lleva a un **tercer requisito**:

- **Definir tipos de datos para cada columna**

La programación orientada a objetos permite, además, crear tipos propios de datos, por ejemplo, los tipos de datos “enumerados”. En este ejemplo se encuentra el tipo de dato “Notas” con 2 valores. ¿Se puede conseguir esto con Excel? Excel tiene sus propios tipos de datos “primitivos”, es decir, números, texto, fecha; pero no es posible definir un tipo de dato aparte.

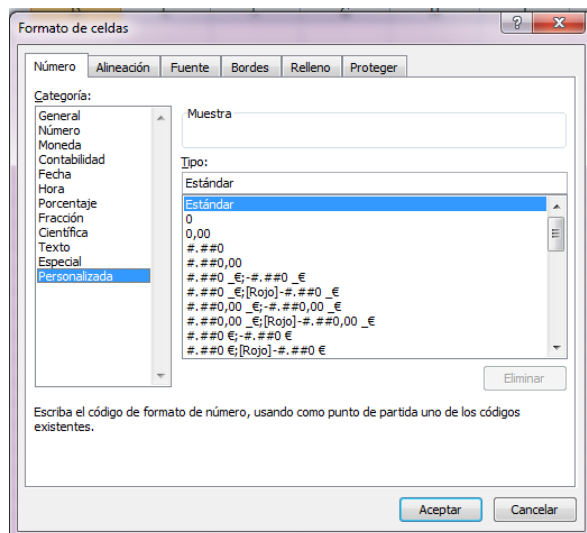


Ilustración 27 - Tipos de datos en excel

Lo que sí permite, como se explicó en el punto 2.2 Funcionalidades, es utilizar la “validación de datos”, donde una de las opciones es introducir los datos en una celda a partir de una lista:

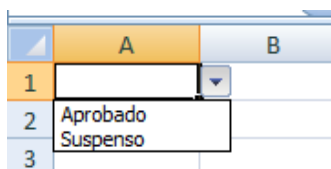


Ilustración 28 - Validación de datos de tipo listado en Excel

La validación de datos vale también para asegurar que los datos que se introducen en una celda

Así pues, sería posible adaptar esta funcionalidad a la definición de tipos de datos del usuario, lo que lleva a la modificación del tercer requisito:

- **Definir tipos de datos para cada columna**
 - o Tipos de datos creados por el usuario

La siguiente cuestión que se plantea es la relación entre las clases. Como se ha visto en el diagrama de clases, la clase examen incluye N objetos alumno. En una hoja de cálculo, ésta referencia se produce entre filas de las diferentes tablas, es decir, **cada objeto de la clase está representado por una fila de la tabla**, por lo tanto se pueden referenciar filas de diferentes clases haciendo uso de una de las mayores utilidades de Excel: creación de referencias entre celdas de otras hojas. Gracias a esto es posible llevar a cabo un **cuarto requisito**:

- **Referencias entre las clases**
 - Permitir introducir objetos dentro de otros objetos.
 - Las referencias deben ser correctas, es decir, no se pueden referenciar objetos de otra clase que no existan.

De forma adicional, ara agrupar todas las clases que se relacionen entre sí, la aplicación permitirá crear **proyectos**, al igual que cuando se programa.`

El resultado de estos 4 requisitos principales es una **hoja de cálculo que cumple con el diseño conceptual de un diagrama de clases**, pero, ¿cómo se genera?

La solución que se propone es utilizar una **herramienta de configuración** de hojas de cálculo que permita su composición mediante un diagrama de clases, pero, como se ha explicado anteriormente, debe ser usable para todos los perfiles de usuario. Para **usuarios programadores**, una aplicación de este tipo es muy **útil** porque se corresponde con la **programación orientada a objetos**, sin embargo una persona que carece de estos conocimientos puede no saber utilizarla. Para que esta herramienta sea “usable” a todos los niveles debe ofrecer **determinadas características que se adapten a diferentes tipos de usuario**. Para ello, una de estas características debe ser **evitar que el usuario básico tenga que programar** macros para el control de los datos. Igualmente debe permitir a los usuarios **expertos** la utilización de columnas y celdas con **fórmulas**, ya que la mayor parte de las hojas de cálculo se utilizan para aplicaciones complejas.

Con esto, queda claro que la herramienta debe adaptarse a diferentes usuarios. Sin embargo, estos usuarios tienen características en común como por ejemplo el uso de formularios en Excel. Muchos de ellos, especialmente los expertos, están acostumbrados a utilizar aplicaciones de formularios en la misma hoja de cálculo, del estilo que se mostró en la Ilustración 10 - Ejemplo aplicación de oficina. Así pues, la **interfaz** de la herramienta debe ser algo parecida, implementada a través de **formularios** y provista de botones, tablas, listas y demás objetos de interfaz iguales a los de Excel. La aplicación, por tanto, será una **herramienta de escritorio**, ya que se pretende implementar una herramienta sencilla accesible por todos los usuarios. Se eliminan por tanto las ideas de servidores Web y otras tecnologías.

Y, ya que se ha generado una hoja con datos correctos, ¿sería posible almacenarlos? Sería útil ofrecer al usuario un sistema de almacenamiento seguro donde guardar los datos sin que dependa de la hoja, ya que ésta, en algún momento dado se puede cerrar o perder. Así pues, surge otro requisito:

- **Almacenamiento de datos correctos**

De forma adicional, las hojas de cálculo se usan para presentar resultados de los datos en diferentes formas de informe, como se vio en el apartado 2.2 Funcionalidades, por lo que el usuario podrá elegir qué tipo de plantilla desea. Puesto que las tablas de Excel pueden ser estáticas o dinámicas y las dinámicas se usan simplemente para una mejor visualización de los datos, la hoja de plantilla que se genere para este tipo de tablas no podrá permitir la carga de datos, por lo que será necesario cargar la hoja de Excel a través de la propia aplicación, de donde surgen dos requisitos más:

- **Carga y almacenamiento de datos**
 - Mediante la aplicación
 - Mediante la hoja de Excel
- **Permitir al usuario elegir tipo de plantilla.**

5. Análisis y Diseño

En este punto se realizará un análisis indicando los requisitos que debe cumplir la aplicación y detallando algunos casos de uso. A continuación se explicará el diseño de la herramienta mediante un esquema general, el diagrama de clases y el diseño de la base de datos.

5.1. Análisis

Como conclusión al punto anterior, se puede realizar un análisis de requisitos que define la aplicación implementada. Este análisis se basa en los requisitos principales deducidos del problema inicial y amplía la funcionalidad de la aplicación.

5.1.1. Especificación de requisitos

En este punto se detallan los requisitos que debe cumplir la aplicación mediante tablas estructuradas de la siguiente forma:

- **ID:** Identificador unívoco del requisito, con la siguiente nomenclatura:
 - RSF_XX: Requisito funcional
 - RSNF_XX: Requisito no funcional
- **Tipo:** subtipo del requisito.
- **Descripción:** explicación del requisito.
- **Importancia:** el dominio de este campo puede tomar los valores: vital, importante o aconsejable.
- **Justificación:** motivo por el cual se ha decidido incluir el requisito.
- **Criterio de cumplimiento:** método para comprobar que el requisito se ha cumplido.

1) Requisitos funcionales

ID	RSF_01
Tipo	Funcional
Descripción	El sistema debe permitir la creación, modificación y eliminación de proyectos. Un proyecto se compondrá de: <ul style="list-style-type: none"> - Nombre - Descripción
Importancia	Vital
Justificación	Sin este requisito no se podrán establecer el diagrama de clases inicial.
Criterio de cumplimiento	Al acceder a la aplicación, lo primero que deberá aparecer será una pantalla de gestión de proyectos.

Tabla 1- Requisito RSF_01

ID	RSF_02
Tipo	Funcional
Descripción	<p>El sistema debe permitir la creación, modificación y eliminación de un diagrama de clases inicial para cada proyecto. Para ello se crearán clases compuestas por:</p> <ul style="list-style-type: none"> - Nombre - Descripción - Hoja de plantilla - Características
Importancia	Vital
Justificación	Este requisito define el diagrama de clases conceptual.
Criterio de cumplimiento	Una vez creado el proyecto (RSF_01), se añadirán clases para el mismo.

Tabla 2 - Requisito RSF_02

ID	RSF_03
Tipo	Funcional
Descripción	<p>Las características (propiedades) de una clase podrán ser creadas, eliminadas y modificadas. Se compondrán de:</p> <ul style="list-style-type: none"> - Nombre - Descripción - Tipo de dato - Fórmula <p>No se podrán gestionar hojas de plantilla en clases sin características.</p>
Importancia	Vital
Justificación	Este requisito define los componentes de una clase.
Criterio de cumplimiento	Una vez creada la clase (RSF_02) se añadirán las características.

Tabla 3 - Requisito RSF_03

ID	RSF_04
Tipo	Funcional
Descripción	<p>La herramienta permitirá crear tipos de datos. Se partirá de tipos de datos primitivos:</p> <ul style="list-style-type: none"> - Número entero positivo y negativo - Número decimal positivo y negativo - Fecha - Texto <p>A partir de estos tipos se podrán crear datos de usuario. A estos datos se les podrán añadir valores, igual que en los tipos de datos “enumerados” correspondientes a un diagrama de clases.</p>
Importancia	Vital
Justificación	Este requisito define el tipo de las características de una clase.
Criterio de cumplimiento	Al crear una característica (RSF_03) se deberá elegir su tipo.

Tabla 4 - Requisito RSF_04

ID	RSF_05
Tipo	Funcional
Descripción	<p>Los tipos de datos serán configurables:</p> <ul style="list-style-type: none"> - Número: <ul style="list-style-type: none"> • Separador de miles y decimales • Moneda • Porcentaje • Rango de números - Texto: <ul style="list-style-type: none"> • Rango de longitud de caracteres
Importancia	Importante
Justificación	La representación de los tipos de dato hace la interfaz más usable para el usuario.
Criterio de cumplimiento	Al crear un tipo de dato se permitirá su configuración.

Tabla 5 - Requisito RSF_05

ID	RSF_06
Tipo	Funcional
Descripción	Las clases tendrán hojas de plantilla. Las hojas de plantilla serán configurables y su configuración se hará a través de diferentes pantallas. Se entiende como hoja de plantilla una hoja de cálculo.
Importancia	Vital
Justificación	Una hoja de plantilla es la representación de una clase en hoja de cálculo.
Criterio de cumplimiento	Cuando se cree una clase, se podrá gestionar su hoja de plantilla.

Tabla 6 - Requisito RSF_06

ID	RSF_07
Tipo	Funcional
Descripción	Los tipos de las hojas de plantilla serán: <ul style="list-style-type: none"> - Tabla estática - Tabla dinámica - Gráfico estático - Gráfico dinámico
Importancia	Importante
Justificación	El tipo de diferentes modelos de plantilla hace referencia a uno de las funcionalidades más importantes y más utilizadas de las hojas de cálculo Excel.
Criterio de cumplimiento	Una de las pantallas de configuración de la hoja de plantilla permitirá escoger su tipo.

Tabla 7 - Requisito RSF_07

ID	RSF_08
Tipo	Funcional
Descripción	Se permitirá escoger qué características son obligatorias, se pueden repetir y pueden llevar subtotal (en caso de que sean numéricas) para la hoja de plantilla.
Importancia	Vital
Justificación	La correcta cumplimentación de las hojas de cálculo es el objetivo de este proyecto. Es necesario especificar las propiedades que deben cumplir los datos que vayan a introducirse para cada clase.
Criterio de cumplimiento	Una de las pantallas de configuración de la hoja de plantilla permitirá configurar las propiedades de cada característica de la clase.

Tabla 8 - Requisito RSF_08

ID	RSF_09
Tipo	Funcional
Descripción	Si el tipo es dinámico, se podrán configurar los siguientes parámetros: <ul style="list-style-type: none"> - Características en filas - Características en columnas - Características en valores - Características en filtro
Importancia	Importante
Justificación	Permite al usuario que la hoja de cálculo ya le salga configurada para así no tener que colocar las características una vez que esté generada.
Criterio de cumplimiento	Una de las pantallas de configuración de la hoja de plantilla permitirá configurar estas propiedades si el tipo es dinámico.

Tabla 9 - Requisito RSF_09

ID	RSF_10
Tipo	Funcional
Descripción	La hoja de plantilla estará unida a una base de datos de seguridad. Se creará una base de datos por proyecto donde cada clase estará representada por una tabla y sus características por columnas de la tabla.
Importancia	Importante
Justificación	Una hoja de cálculo puede actuar como elemento de visualización de datos o como una pequeña base de datos. Se ofrece al usuario la posibilidad de almacenar estos datos sin depender únicamente de la hoja de datos que se genere. La hoja de datos se generará con los datos almacenados
Criterio de cumplimiento	Al crear un proyecto se creará una base de datos de seguridad. Al crear una clase se creará una tabla en esa base de datos. Cuando se configure una característica se modificará la tabla añadiendo una columna más.

Tabla 10 - Requisito RSF_10

ID	RSF_11
Tipo	Funcional
Descripción	La hoja de plantilla se podrá cargar mediante la aplicación. Se generará una tabla provista de tantas columnas como características tenga la clase y con todos los métodos de validación que cumplan con las propiedades del tipo de dato. Soportará y validará las referencias entre las clases.
Importancia	Importante
Justificación	Haciendo referencia al requisito RSF_10, se permitirá al usuario una primera carga mediante una tabla de generada a través de la aplicación.
Criterio de cumplimiento	Una vez configurada toda la hoja de plantilla, se podrá cargar la hoja.

Tabla 11 - Requisito RSF_11

ID	RSF_12
Tipo	Funcional
Descripción	<p>Se generará la hoja de cálculo correcta y se permitirá su carga a través de un botón incluido en la misma hoja.</p> <p>Si la hoja es una tabla estática:</p> <ul style="list-style-type: none"> - Se generará una tabla Excel compuesta por tantas columnas como características. - La hoja estará controlada por macros que aseguran la calidad de los datos y métodos de validación. - El código se generará de forma automática. <p>Si es una tabla dinámica, gráfico estático o gráfico dinámico:</p> <ul style="list-style-type: none"> - No se permitirá carga y será una hoja de cálculo para visualización de datos. Esta hoja deberá haberse cargado previamente desde la aplicación.
Importancia	Vital
Justificación	El objetivo final de este proyecto es generar una hoja de cálculo correcta. Este requisito es el que finaliza este objetivo. La hoja de cálculo se generará con todos los controles necesarios y además permitirá la carga.
Criterio de cumplimiento	Una vez configurada toda la hoja de plantilla, se podrá generar la hoja de cálculo Excel de forma automática con todo el código de las macros incluido.

Tabla 12 - Requisito RSF_12

ID	RSF_13
Tipo	Funcional
Descripción	<p>Cada fila de la hoja de plantilla corresponderá a un objeto, por lo que las referencias entre las clases que contengan objetos de otra deberán ser válidas. Es decir, el sistema deberá comprobar que se está referenciando a un objeto que existe la clase que contiene.</p> <p>De la misma forma, no se podrá modificar un objeto que esté referenciado.</p>
Importancia	Vital
Justificación	Las referencias entre los datos deben ser correctas.
Criterio de cumplimiento	No se podrán insertar datos que no existan en la clase referenciada. Tampoco se podrán modificar los objetos cuyos datos estén referenciados.

Tabla 13 - Requisito RSF_13

ID	RSF_14
Tipo	Funcional
Descripción	No será posible incluir referencias de dos columnas (Atributos) del mismo objeto en una clase, ya que, ambos datos tendrían que coincidir y esto puede llevar a errores.
Importancia	Importante
Justificación	Las referencias entre los datos deben ser correctas.
Criterio de cumplimiento	En la configuración de la hoja no se permitirá añadir varias referencias de un mismo objeto.

Tabla 14 - Requisito RSF_14

2) Requisitos no funcionales:

ID	RSNF_01
Tipo	No funcional - Usabilidad
Descripción	La interfaz estará provista de formularios similares a los de Excel.
Importancia	Vital
Justificación	La interfaz debe ser usable para todos los tipos de usuario de Excel, acostumbrados a trabajar con objetos gráficos de las hojas de cálculo.
Criterio de cumplimiento	Evitar uso de botones con formas y colores diferentes a los de los formularios en Excel. Evitar objetos de interfaz gráfica y pantallas desconocidos para este tipo de usuarios.

Tabla 15 - Requisito RSNF_01

ID	RSNF_02
Tipo	No funcional - Usabilidad
Descripción	La aplicación estará preparada para soportar tres tipos de usuario: <ul style="list-style-type: none"> - Básico: sin muchos conocimientos en hojas de cálculo. - Intermedio: capaz de interactuar con algunas funcionalidades en Excel, tales como filtros y fórmulas. - Usuario experto: capaz de programar y entender las referencias entre hojas, fórmulas más complejas y validaciones de datos.
Importancia	Vital
Justificación	Es importante que todos los tipos de usuario de las hojas de cálculo Excel se atrevan a utilizar este tipo de aplicación para ayudarse a generar hojas de cálculo correctas.
Criterio de cumplimiento	La hoja se generará con filtros, se permitirá el uso de fórmulas que se calculan automáticamente y tendrá validaciones de datos. Funcionalidades que son comunes en los usuarios de Excel.

Tabla 16 - Requisito RSNF_02

ID	RSNF 03
Tipo	No funcional - Portabilidad
Descripción	<p>La aplicación se ajustará a un diseño de tres capas:</p> <ul style="list-style-type: none"> - Controlador de base de datos y Excel <ul style="list-style-type: none"> • Independiente de las otras capas para permitir el cambio de base de datos. - Capa intermedia: <p>Adaptada al lenguaje de consultas de la base de datos del controlado.</p> - Capa interfaz: <ul style="list-style-type: none"> • Formularios independientes de las capas anteriores permitiendo la portabilidad a web.
Importancia	Importante
Justificación	Si se realiza un buen diseño por capas, se permitirá la posible ampliación de la aplicación, pudiendo hacerse en web o con bases de datos más grandes.
Criterio de cumplimiento	Las capas tendrán la dependencia descrita.

Tabla 17 - Requisito RSFN_03

ID	RSNF 01
Tipo	No funcional
Descripción	Toda la configuración de los proyectos se almacenará en una base de datos. La aplicación trabajará sobre ella.
Importancia	Vital
Justificación	La interfaz debe ser usable para todos los tipos de usuario de Excel, acostumbrados a trabajar con objetos gráficos de las hojas de cálculo.
Criterio de cumplimiento	Evitar uso de botones con formas y colores diferentes a los de los formularios en Excel. Evitar objetos de interfaz gráfica y pantallas desconocidos para este tipo de usuarios.

5.1.2. Casos de uso

A continuación se muestra el diagrama definitivo de los casos de uso junto con algunas descripciones de las acciones más importantes que puede realizar el usuario:

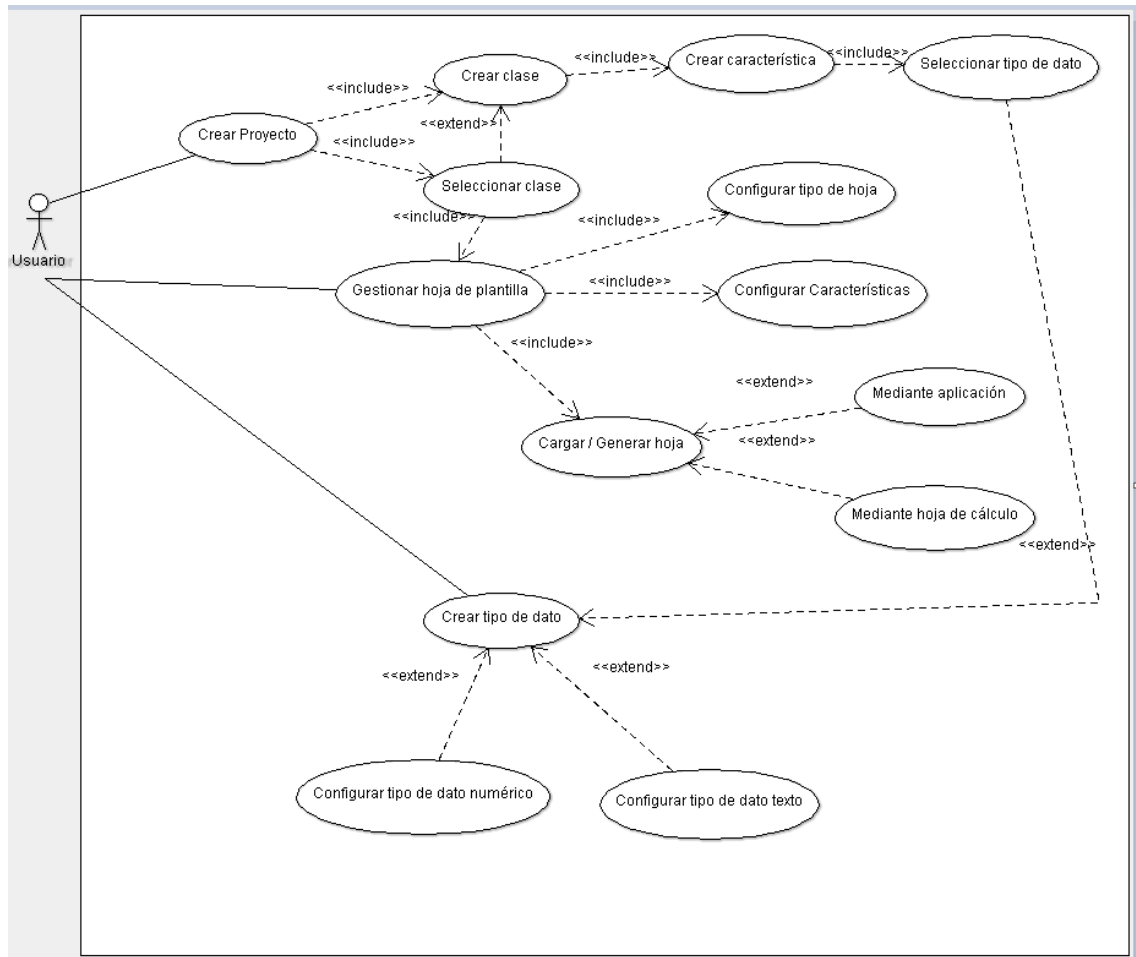


Ilustración 29 - Diagrama de casos de uso

5.1.2.1. Caso de uso: carga de la hoja de plantilla mediante aplicación

Actor Primario:

Usuario

Interesados y Objetivos:

El usuario desea cargar la hoja de plantilla que ha configurado a través de la aplicación.

Precondiciones:

La hoja de plantilla debe haberse gestionado y configurado previamente.

Garantía de éxito (Postcondiciones):

Se almacena en la base de datos de seguridad los datos que se introduzcan correctamente en la tabla de datos de la aplicación.

Escenario Principal de Éxito:

1. El usuario ha configurado las características que debe tener la plantilla y accede a la pantalla de carga para comenzar la inserción de los datos.
2. En la pantalla de carga se muestra una tabla que permite edición en cada una de sus celdas.
3. La tabla tendrá las siguientes propiedades:
 - Las columnas serán las características de la clase.
 - El tipo de la columna se corresponderá al tipo configurado para la característica.
 - I. Se mostrará en el formato que se haya especificado.
4. Cuando el usuario finalice la edición, la aplicación permitirá guardar los datos en la copia de seguridad.
5. Si se vuelve a entrar en la aplicación, la herramienta cargará automáticamente estos datos.

Extensiones (Flujos alternativos):

4. Se introduce un dato que no cumple con las propiedades del tipo de dato para esa columna.
 - a. Se muestra un mensaje de error que indica que el dato es inválido. Se cancela la edición para esa celda obligando al usuario a insertar un dato correcto.
 - b. Si el valor para esa característica es obligatorio y el valor introducido es blanco, se cancela la edición y se obliga al usuario a insertar un dato.
 - c. Si no se permiten valores repetidos para esa característica, se muestra un mensaje de aviso y se cancela la edición.
5. El sistema falla cuando se está guardando la copia de seguridad:
 - a. Se mostrará un mensaje de error indicando el fallo.

Requisitos Especiales:

Respuesta rápida de la herramienta si se produce un fallo al guardar en la base de datos de seguridad.

Lista de variaciones de tecnología y datos:

Se puede reintentar la posibilidad de guardar en la base de datos de seguridad.

Frecuencia de ocurrencia:

Alta, dado que el usuario cargará las hojas para después visualizarlas en Excel.

Temas abiertos:

Se propone mejorar la aplicación para el uso de métodos eficientes de validación de datos.

**5.1.2.2. Caso de uso: carga de la hoja de plantilla mediante hoja de cálculo en Excel /
Generación de hoja de cálculo*****Actor Primario:***

Usuario

Interesados y Objetivos:

El usuario desea cargar y/o generar la hoja de cálculo de Excel.

Precondiciones:

La hoja de plantilla debe haberse gestionado y configurado previamente.

Garantía de éxito (Postcondiciones):

Se genera una hoja de cálculo correcta a partir de la configuración especificada en la hoja de plantilla.

Escenario Principal de Éxito:

1. Una vez que el usuario haya terminado de configurar la hoja de plantilla para la clase que ha seleccionado, accederá a la pantalla de carga.
2. Esta pantalla permitirá, o bien cargar desde la aplicación o bien exportar la hoja a Excel.
 - El usuario elegirá la segunda opción.
3. Se generará una hoja de cálculo en Excel provista de:
 - Una tabla que coincide con la configuración de la hoja de plantilla.
 - I. Permitirá la inserción y modificación de los datos.
 - Macros que controlan la validación de los datos.
4. Una vez que haya finalizado la carga, la hoja de cálculo permitirá el almacenamiento en Excel.

Extensiones (Flujos alternativos):

6. Se introduce un dato que no cumple con las propiedades del tipo de dato para esa columna.
 - a. Se muestra un mensaje de error que indica que el dato es inválido. Se cancela la edición para esa celda obligando al usuario a insertar un dato correcto.
 - b. Si el valor para esa característica es obligatorio y el valor introducido es blanco, se cancela la edición y se obliga al usuario a insertar un dato.
 - c. Si no se permiten valores repetidos para esa característica, se muestra un mensaje de aviso y se cancela la edición.
7. El sistema falla cuando se está guardando la copia de seguridad:
 - a. Se mostrará un mensaje de error indicando el fallo.

Requisitos Especiales:

Respuesta rápida de la herramienta si se produce un fallo al guardar en la base de datos de seguridad.

Lista de variaciones de tecnología y datos:

Se puede reintentar la posibilidad de guardar en la base de datos de seguridad.

Frecuencia de ocurrencia:

Alta, dado que el usuario cargará las hojas para después visualizarlas en Excel.

Temas abiertos:

Se propone mejorar la aplicación para el uso de métodos eficientes de validación de datos.

5.2. Diseño

5.2.1. Esquema general

El esquema general de la aplicación se muestra en la siguiente imagen:

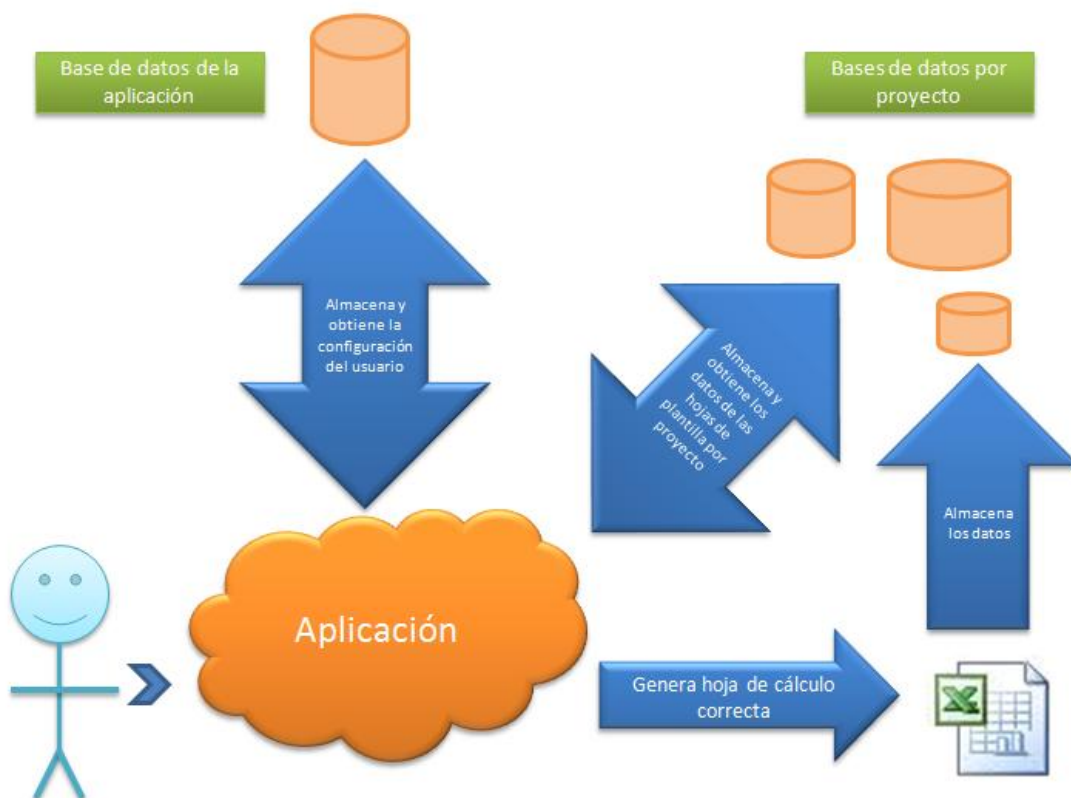


Ilustración 30 - Esquema general del diseño

Como se puede observar, el usuario, a través de la aplicación puede generar una hoja de cálculo. La aplicación es la encargada de, por un lado, almacenar y obtener la información que ha configurado el usuario:

- Proyectos
 - o Diagramas de clases
- Tipos de datos
- Configuración de hojas de plantilla

Este almacenamiento se realiza mediante la **base de datos de la propia aplicación**.

Por otro lado, se generarán **bases de datos de seguridad** donde almacenar los datos de los proyectos. Estos datos se corresponden a los objetos de las clases, es decir, los datos que el usuario introduce en las tablas.

La aplicación además, construye una hoja de cálculo correcta donde se genera **código automático**, es decir, las macros que controlan la hoja.

5.2.2. Diseño de la aplicación

5.2.2.1. Arquitectura de tres capas

Uno de los requisitos de la aplicación (Tabla 17 - Requisito RSFN_03) es que la aplicación tenga un diseño de tres capas. Estas capas son:

- **Capa BD:** es la encargada de interactuar con la base de datos, tanto de seguridad como de la aplicación. Esta capa tendrá una clase cuyos objetivos son tramitar la conexión a las bases de datos, realizar consultas, etc.
Tendrá otra clase, además, que manejará las librerías y objetos de Excel. Esta clase será la que cree el objeto “libro”, maneje las hojas y escriba las macros para la hoja de cálculo.
Esta capa contiene todas las clases de código más complejo y de más bajo nivel.
- **Capa Interfaz_Objeto_BD:** capa intermedia que contiene las diferentes clases que interactúan con los objetos de la “Capa_BD”. Las clases de esta capa son las encargadas de escribir las consultas y realizar las llamadas de ejecución a través de la capa BD. En esta capa se encontrarán las clases análogas a las entidades de la base de datos.
- **Capa Interfaz:** contiene todos los formularios de la aplicación. Estos formularios interactuarán e intercambiarán objetos de la capa “Interfaz_Objeto_BD” y se encargarán de mostrar los resultados de las consultas y ofrecer al usuario la posibilidad de interactuar con la base de datos mediante una interfaz usable.

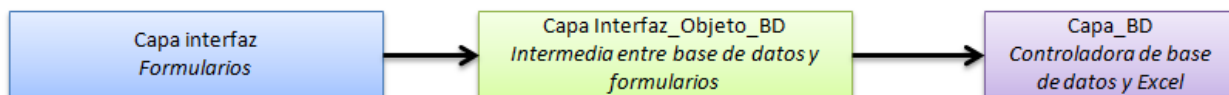


Ilustración 31 - Esquema básico diseño tres capas

En la siguiente imagen se puede ver cómo interactúan las capas entre sí de forma esquemática:

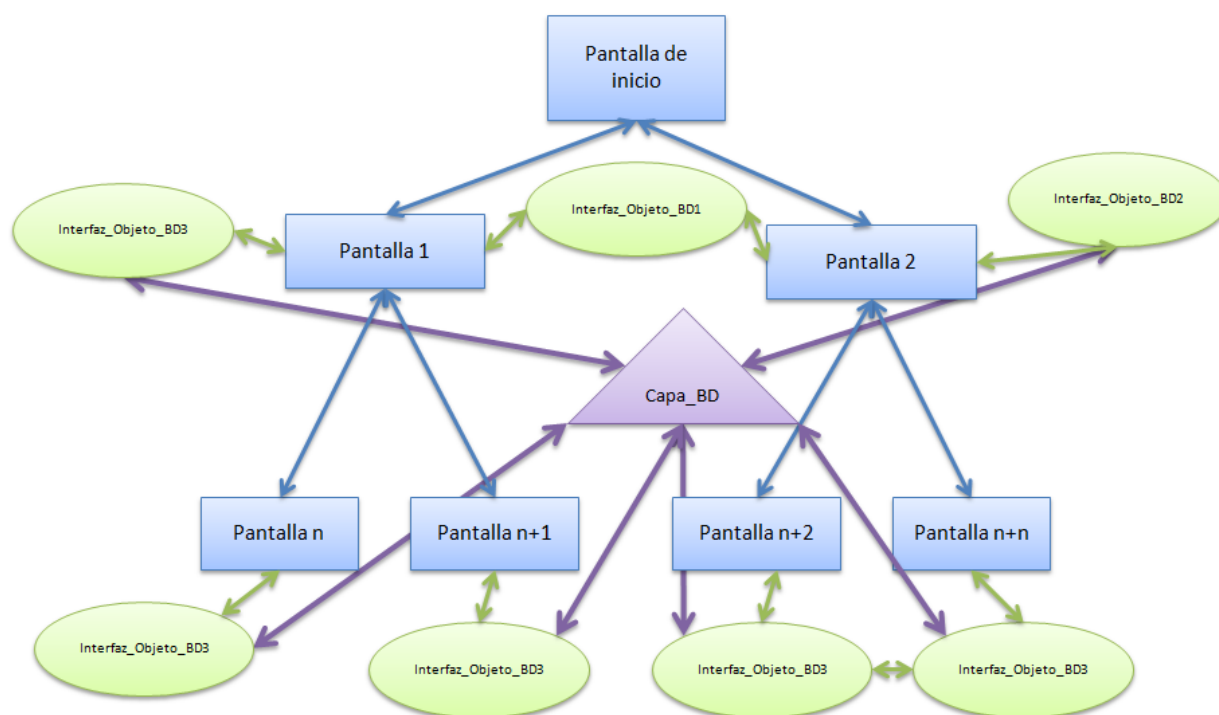


Ilustración 32 - Esquema interacción entre capas

Siguiendo el sistema de colores de la imagen Ilustración 31 - Esquema básico diseño tres capas, se observa cómo las pantallas (formularios) se relacionan con los objetos de la capa “interfaz_objeto_bd”. Esta relación es bidireccional porque la pantalla obtiene información de la capa interfaz pero a su vez pasa información a través de ella. Se puede observar también que las pantallas pueden compartir objetos de esta capa y que además los objetos pueden estar relacionados entre sí.

Los objetos de la capa “interfaz_objeto_bd” interactúan a su vez con objetos de la “capa_bd”. De la misma forma que anteriormente, la relación es bidireccional.

5.2.2.1.1. *Ventajas de la arquitectura de tres capas*

Este diseño es un diseño eficiente ya que permite, entre otras cosas, la independencia de las capas. Lo que se consigue es que la portabilidad de la aplicación con respecto a la base de datos, web o librerías de Excel sea más sencilla, ya que sólo sería necesario modificar clases concretas de una capa.

En el caso de querer cambiar de gestor de base de datos, con modificar el conector de la “capa_bd” sería suficiente. En caso de que el lenguaje fuera muy diferente, podría darse el caso de tener que retocar objetos de la “capa_interfaz_bd”. Dada esta situación, podrían tenerse diferentes clases en función del gestor que se utilice sin tener que retocar código de otras capas.

En el caso de la portabilidad a Web, ocurre lo mismo. Si el gestor no se modifica tan sólo sería necesario modificar la parte de la interfaz de los formularios.

5.2.3. Diagrama de clases

Atendiendo a la arquitectura de tres capas, se crearán tres paquetes diferentes:

- Capa_BD: formada por las clases de más bajo nivel. Serán las que interactúen con la base de datos y las librerías de Excel.
- Capa_Objeto_BD: contendrá las clases que actuarán de mediadoras entre la capa baja y la capa interfaz.
- Capa_Interfaz: contendrán los formularios y las clases de interfaz gráfica.

Debido al extenso diagrama, su explicación se dividirá en las diferentes capas. Las imágenes correspondientes a cada una de las clases se pueden encontrar en los anexos.

5.2.3.1. Capa_BD

Contiene las clases:

- UtilsBD: encargada de conectar a la base de datos y ejecutar consultas.
- ManejadorExcel: encargada de interactuar con las hojas de cálculo y manejar las librerías necesarias.

Las imágenes de estas clases se pueden ver en los anexos Clases de la Capa_BD.

5.2.3.2. Capa_Objeto_BD

Formada por las siguientes clases:

- Característica: representa la entidad “Característica” de la base de datos.
- Características: realiza consultas a la base de datos y gestiona, en forma de tablas de datos, los resultados de las mismas.
- Clase: representa la entidad “Clase” de la base de datos.
- Clases: interactúa con la base de datos por medio de tablas de datos con la información de las clases.
- HojaPlantilla: representa la entidad “HojaPlantilla” de la base de datos.
- HojasPlantillas: contiene la información de la configuración de la hoja de plantilla de la clase gestionada por medio de tablas de datos.
- Proyecto: representa la entidad “Proyecto” de la base de datos.
- Proyectos: contiene la información de los proyectos de la base de datos en forma de tabla.
- TipoDato: representa la entidad “TipoDato” de la base de datos.
- TiposDatos: interactúa con la base de datos para obtener la información de TipoDato mediante tablas de datos.
- TipoPlantilla : representa la entidad TipoPlantilla de la base de datos.

Las imágenes de estas clases se pueden ver en los anexos Clases Capa_Objeto_BD.

5.2.3.3. Capa Interfaz

Formada por las clases que gestionan los formularios. Cada una de estas clases hace referencia a una pantalla. El conjunto de clases que se genera es doble ya que por un lado está la clase formulario correspondiente al diseño y por otro lado la parte correspondiente al código, por lo que es necesario contar las dos.

En la siguiente imagen se muestra el listado de las clases de la capa interfaz. La relación entre cada una de las clases y su información se encuentra detallada en los anexos de Capa Interfaz.

+	AltaModDatos
+	AltaModDatos
+	AltaModificacionCaracteristica
+	AltaModificacionCaracteristica
+	AltaModificacionProyectos
+	AltaModificacionProyectos
+	AnadirObjetoClase
+	AnadirObjetoClase
+	AnadirValores
+	AnadirValores
+	CargarHojaAplicacion
+	CargarHojaAplicacion
+	ConfigTipoDatoFecha
+	ConfigTipoDatoFecha
+	ConfigTipoDatoNumero
+	ConfigTipoDatoNumero
+	ConfigTipoDatoTexto
+	ConfigTipoDatoTexto
+	ConfiguracionDatos
+	ConfiguracionDatos
+	ConfiguracionFormulaCaracteristica
+	ConfiguracionFormulaCaracteristica
+	GestionProyectos
+	GestionProyectos
+	GestionarClases
+	GestionarClases
+	GestionarHojaPlantillaCaracteristicas
+	GestionarHojaPlantillaCaracteristicas
+	GestionarHojaPlantillaDinamica
+	GestionarHojaPlantillaDinamica
+	GestionarHojaPlantillaMenu
+	GestionarHojaPlantillaMenu
+	GestionarHojaPlantillaObjetoContenido
+	GestionarHojaPlantillaObjetoContenido
+	GestionarHojaPlantillaTipo
+	GestionarHojaPlantillaTipo
+	MenuInicio
+	MenuInicio
+	Presentacion
+	Presentacion
+	TipoGrafico
+	TipoGrafico

5.2.4. Esquema de la base de datos

La base de datos de la aplicación se compone de los siguientes elementos principales:

Nombre	Proyecto
Descripción	Tabla donde se almacenan los proyectos
Atributos	<ul style="list-style-type: none"> - Id - Nombre - Descripción

Tabla 18- BD - Entidad "Proyecto"

Nombre	Objeto
Descripción	Tabla donde se almacenan las clases.
Atributos	<ul style="list-style-type: none"> - Id - Nombre - Descripción - Id del proyecto

Tabla 19- BD- Entidad "Objeto"

Nombre	Característica
Descripción	Representa el elemento conceptual "característica" de la clase.
Atributos	<ul style="list-style-type: none"> - Id - Nombre - Descripción - idTipoDato - idObjeto: clase a la que pertenece

Tabla 20 - BD - Entidad "Característica"

Nombre	TipoDato
Descripción	Tabla donde se almacenan los tipos de dato que puede tener una característica
Atributos	<ul style="list-style-type: none"> - Id - Nombre - Descripción - idTipoDatoPadre: identificador que puede ser: <ul style="list-style-type: none"> • 1: número entero • 2: número decimal • 3: Texto • 4: Fecha

Tabla 21 – BD - Entidad "TipoDato"

Nombre	ValorTipoDato
Descripción	Tabla donde se almacenan los valores correspondientes a los tipos de datos (De igual forma que los enumerados).
Atributos	<ul style="list-style-type: none"> - Id - Nombre: valor - idTipoDato

Tabla 22 - BD - Entidad "TipoDato"

Nombre	TipoPlantilla
Descripción	Tabla donde se guardan los tipos de hojas que se pueden configurar.
Atributos	<ul style="list-style-type: none"> - Id - Nombre <ul style="list-style-type: none"> • Tabla estática • Tabla Dinámica • Gráfico Dinámico • Gráfico estático

Tabla 23 - BD - Entidad "TipoPlantilla"

Las tablas que representan **relaciones** son:

- HojaPlantillaObjeto: une el identificador del objeto (clase) con el identificador de la plantilla. Es coincidente el identificador del objeto (clase) con el identificador de la hoja de plantilla, por eso no es necesaria una tabla de hojas de plantilla.
- HojaPlantillaTipoPlantilla: une el identificador de la hoja de plantilla con el identificador del tipo de plantilla.
- HojaPlantillaConfiguracionDinamica: guarda el identificador de la hoja con su configuración dinámica:
 - Id: identificador de la hoja
 - idCaracteristica
 - posición
 - FT: Filtro
 - F: fila
 - V: valor
 - C: columna

- HojaPlantillaConfiguracionEstatica: guarda el identificador de la hoja con su configuración estática:
 - idHojaPlantilla
 - idCaracteristica
 - obligatoria
 - repetida
 - subtotalLos 3 últimos son valores booleanos que indican si se permiten valores repetidos, si se quiere el subtotal en la hoja de Excel o si se desea que el valor sea obligatorio.
- ObjetosContenidos: representa las relaciones entre las clases. Guarda el identificador de la clase contenedora y el de la clase contenida.
- TipoDatoNumero: almacena la configuración de los tipos de datos numéricos.
- TipoDatoTexto: almacena la configuración de los tipos de datos texto.

El esquema relacional de la base de datos se muestra en la siguiente imagen:

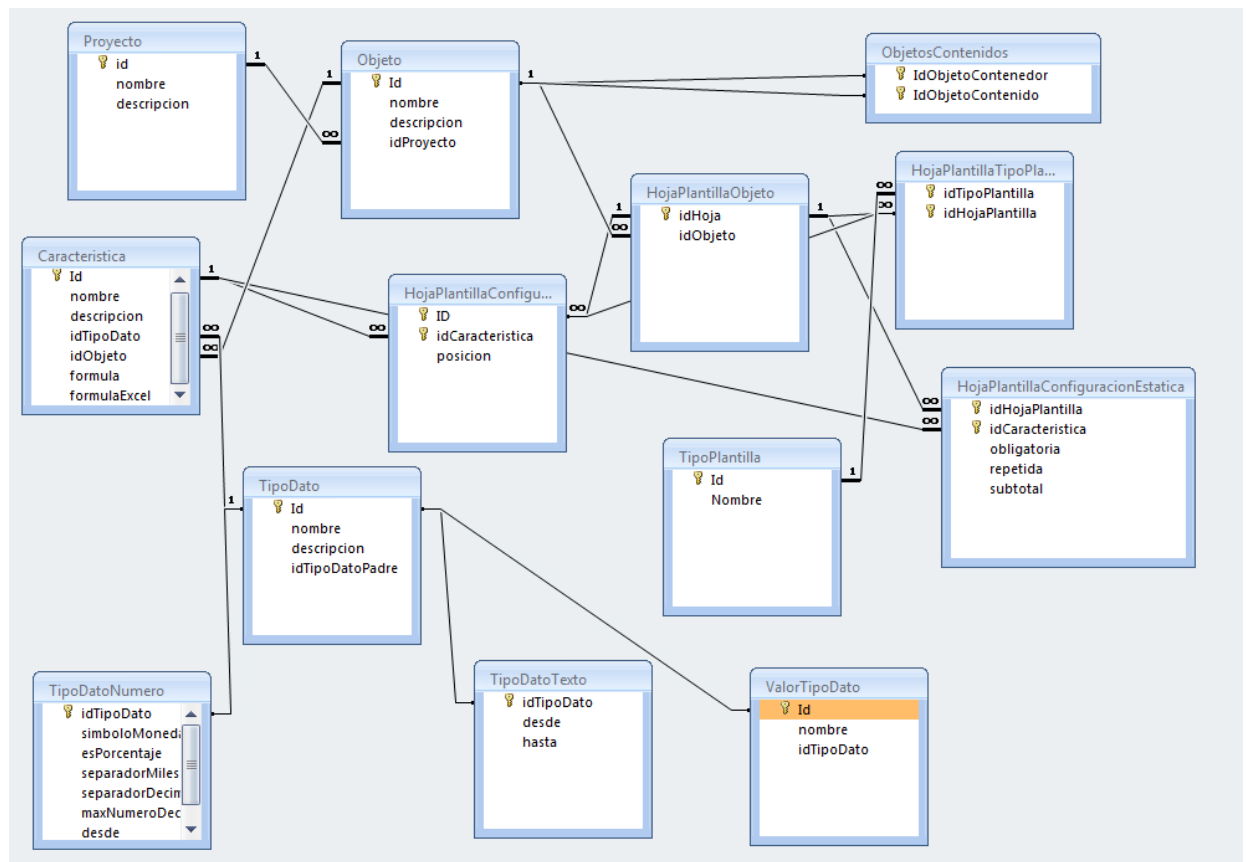


Ilustración 33- Esquema relacional BD

6. Implementación

En este punto se explicará de qué forma se ha desarrollado la aplicación. En primer lugar se estudiarán las alternativas tecnológicas elegidas y a continuación se explicarán los puntos más importantes de su implementación.

6.1. Plataforma .NET frente a otras alternativas

Ya que es necesario generar y controlar hojas de Excel mediante macros, la primera opción planteada es desarrollar la herramienta mediante **formularios en VBA**:

- Ventajas:
 - Interfaz gráfica sencilla de implementar
 - Los usuarios de Excel están acostumbrados a utilizar este tipo de formularios.
 - Permite generar y controlar las hojas directamente.
- Desventajas
 - No es un lenguaje orientado a objetos.
 - Dificultad de implementación.
 - No permite portabilidad
 - Impide reutilización de código.
 - No permitiría trasladarse a web
 - Conexión a base de datos
 - Tan sólo para bases de datos Microsoft.
 - Librerías y tecnologías limitadas.

Debido a sus importantes desventajas, se planteó una segunda opción: **lenguajes orientados a objetos**, como por ejemplo **Java**:

- Ventajas:
 - Facilidad de implementación debido a los conocimientos de la desarrolladora.
 - Permite portabilidad
 - A Web
 - Reutilización de código.
 - Diseño de 3 capas.
- Desventajas:
 - Las librerías que controlan Excel no son oficiales.
 - El código se debe adaptar a la propia herramienta.
 - El código es complejo de manejar.

Por las ventajas, queda claro que la opción más correcta es usar un lenguaje orientado a objetos que permita un manejo sencillo de las hojas de cálculo en Excel. La solución que mejor se adapta a esta situación es el uso de la **plataforma .NET**.

Dentro de la plataforma .NET se puede programar en:

- C#
- VB.NET
- ASP : orientado a web

Puesto que VB.NET tiene la misma sintaxis que VBA, lo más fácil para generar el código automático de las hojas de cálculo es la utilización de **VB.NET** por lo que ha sido la opción escogida.

Otra de las razones del uso de .NET frente a otros lenguajes orientados a objetos es que los formularios en esta plataforma están divididos en 2 ficheros:

- Código .vb (o .cs en el caso de C#)
- Interfaz del formulario

El código .vb o .cs se puede modificar sin tener que modificar el diseño y viceversa. Esto permite que, si por ejemplo se quiere trasladar a web, sólo sería necesario modificar la interfaz del formulario y adaptarla al lenguaje “front” que se utilice. En el caso de esta plataforma, el lenguaje web sería ASP que se acomoda de manera automática al código que se ejecuta por detrás, lo que supone un ahorro de tiempo a los programadores.

La IDE utilizada ha sido **Visual Studio 2010** debido a que es una herramienta Microsoft recomendada para esta plataforma.

6.2. Base de datos Access frente a otras alternativas

Puesto que se decidió implementar la herramienta en la plataforma .NET, perteneciente a Microsoft y unida a las hojas de cálculo en Excel, se tomó como primera alternativa las **bases de datos en Access** por la **sintonía con la tecnología utilizada**.

Según (12), es recomendable el uso de Access para usuarios personales y pequeñas empresas ya que es una herramienta sencilla y más simple que otras tecnologías.

Se entiende que los usuarios de Excel tienen licencias para Microsoft Office, por lo que no tendrían problemas en el uso de bases de datos en Access. El caso contrario ocurre en:

- **Sql Server**
- **Oracle**

Estas dos herramientas, a pesar de ser mucho más robustas y orientadas a ambientes más grandes, tienen **licencias de alto coste**, lo que evita utilizarlas.

Otra alternativa fue MYSQL, que, a pesar de ser gratuita y más eficiente que Access, es un gestor más complejo.

Debido al conocimiento y experiencia laboral de la desarrolladora en bases de datos Access, se tomó como decisión final incorporar esta base de datos a la aplicación.

6.3. Implementación

Cumpliendo con los requisitos y el diseño de la aplicación, el código está dividido en 3 capas:

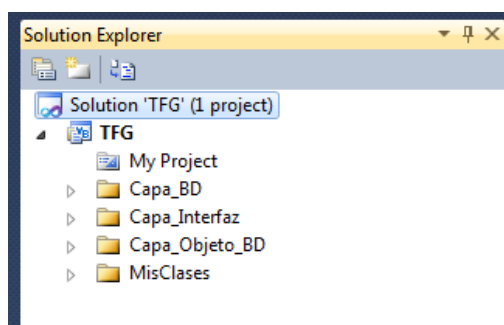


Ilustración 34 - Solution Explorer aplicación

En esta imagen se observa el explorador de soluciones del proyecto, compuesto por:

- El proyecto TFG
 - Dividido en paquetes:
 - Capa_BD: capa baja
 - Capa_Interfaz: capa alta
 - Capa_Objeto_BD: capa intermedia
 - MisClases: paquete que contiene clases auxiliares
 - Cada uno de estos paquetes contiene las clases de las que se habló en el diagrama de clases.

La Capa_BD y la Capa_Objeto_BD incluyen ficheros en .vb, es decir, ficheros en código Visual Basic. Estos ficheros contienen las clases con sus atributos y propiedades.

A continuación se explicará la implementación de las clases más importantes:

6.3.1. Clase Utils BD

Esta clase pertenece al paquete CAPA_BD y es la encargada de gestionar las conexiones a la base de datos. Para ello, utiliza el proveedor de datos .NET Framework **OleDb**.

Este proveedor ofrece una colección de clases que permite acceder a un origen de datos en concreto y rellenar objetos mediante **DataSet** con los datos obtenidos de la base de datos.

6.3.2. Clase ManejadorExcel

Esta clase también se encuentra en la “Capa_BD” y es la encargada de interactuar con las hojas de cálculo. Para ello se utiliza la librería **Interoperable**, perteneciente también a la plataforma .NET.

Esta librería permite crear objetos que manejan la aplicación Excel, el libro Excel y la hoja Excel, lo que quiere decir que se puede generar una hoja de cálculo y escribir en ella mientras la aplicación se está ejecutando.

La clase “ManejadorExcel” es la encargada de pintar las tablas de datos en las hojas, las macros de validación de datos y de crear los tipos diferentes de informe a partir de las plantillas que se han configurado.

Esta librería tiene como objetivo, además, generar el código automático de las hojas de cálculo, es decir, las **macros**. Para ello, accede al proyecto VBA y recorre todos sus componentes, añadiendo las macros donde corresponda. Se encarga, además, de ejecutar las macros a través del método .Run. Una ejecución de una macro es, en realidad, una llamada a una función que se ejecuta sobre la aplicación de Excel.

Las macros son cadenas de código que se van escribiendo utilizando la serie de instrucciones anteriores.

El resultado final de lo que realiza esta clase es el código de la macro escrita en el módulo de Excel y ejecutándose mientras Excel está abierto:

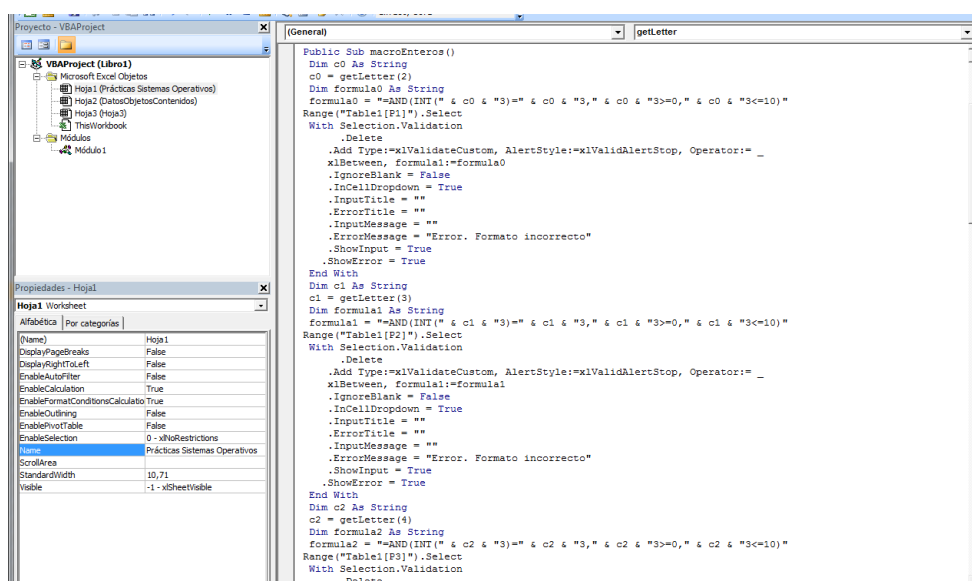


Ilustración 35 - Resultado final macro

Cuando se genera la hoja de cálculo, se escribe en una hoja oculta los datos de referencia del objeto contenido o contenedor, es decir, que si una clase contiene un objeto se escribe en una hoja de Excel los datos de la columna del objeto al que está referenciado (sin valores repetidos) y de esta forma, para saber si el dato existe, se busca en esa hoja oculta. Es un método rápido y eficaz debido a los métodos de búsqueda de las hojas de cálculo.

Igualmente, si se está modificando una clase que tiene referencias de otras clases, no se podrán modificar los valores de las columnas que están referenciadas. Esos datos se escriben en otra hoja oculta y se busca en ella de la misma forma que anteriormente.

6.3.3. Formularios – Interfaz gráfica

El paquete “Capa_Interfaz” está compuesto por los formularios de la aplicación. Los formularios son las pantallas de interacción con el usuario, formados por dos archivos:

- **Designer:** hace referencia a la interfaz del formulario. Es el fichero resultante de introducir elementos gráficos en la ventana:

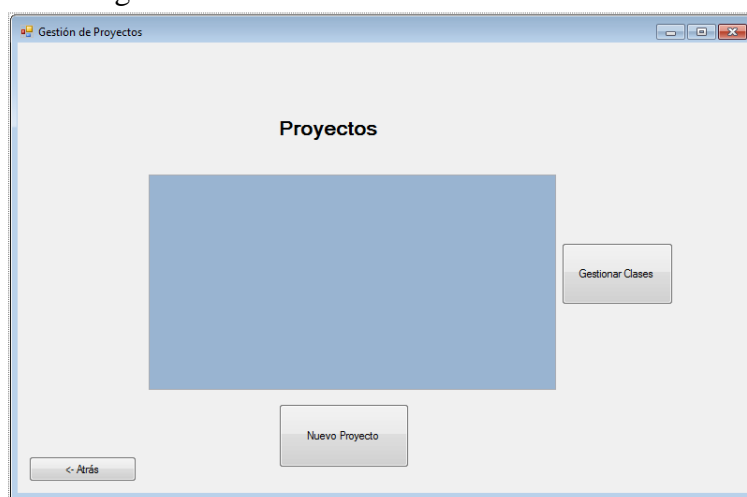


Ilustración 36 – Designer

- **Code:** es la parte del formulario que hace referencia al código que se ejecuta por detrás. Este código está formado por eventos onLoad, onChange, onClick, etc. de los elementos gráficos del designer.

En la parte de code se hacen llamadas a los métodos definidos en las clases de la “capa_objeto_bd”. En la siguiente imagen se muestra un ejemplo de lo que ocurre al borrar un proyecto. Este método es el evento “onUserRowDeleting” que se ejecuta cuando el usuario pulsa el botón suprimir sobre una tabla:

```
Private Sub UserDeletingRow(ByVal sender As Object, _  
    ByVal e As DataGridViewRowCancelEventArgs) _  
    Handles DataGridView1.UserDeletingRow  
  
    Dim result = MessageBox.Show("¿Estás seguro de que quieres borrar el proyecto?", "Aviso", MessageBoxButtons.YesNo)  
    If result = DialogResult.No Then  
        e.Cancel = True  
    ElseIf result = DialogResult.Yes Then  
  
        p.borraProyecto(Convert.ToInt32(e.Row.Cells(0).Value.ToString), e.Row.Cells(1).Value.ToString)  
    End If  
End Sub
```

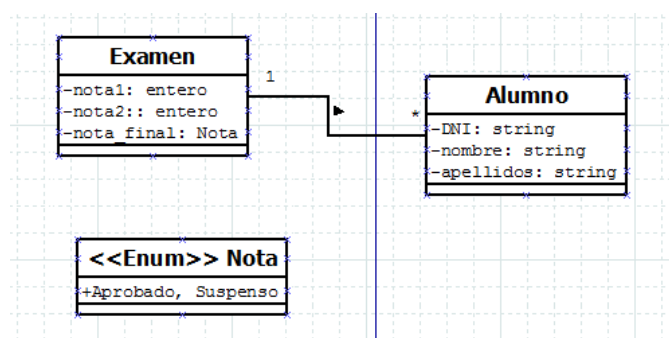
Ilustración 37 - Code

7. Pruebas y casos de estudio

En este punto se van a estudiar algunos ejemplos concretos de posibles usos de la aplicación. Para ello se partirá del problema inicial que se planteó en la explicación de la solución propuesta y después se estudiarán algunos casos adaptados a otros tipos de usuarios.

7.1. Problema inicial

El primer caso de prueba toma como base el punto 4.1 Análisis del problema inicial, donde, se quiere generar una hoja de Excel donde poder almacenar las calificaciones de los alumnos de una asignatura, asegurando que no se inserten alumnos con un formato incorrecto y que las notas sean números del 1 al 10. El diagrama de clases del que se parte es el siguiente:



Como se verá a continuación a partir de este diagrama se ha creado uno algo más complejo donde se modifican los **tipos de datos de DNI, nota1 y nota2** para ver cómo crear una hoja de cálculo correcta a partir de esta situación.

A parte, a la clase **Examen** se le ha añadido un nuevo atributo, **“nota_media”**, calculada como $(\text{nota1} + \text{nota2}) / 2$.

En primer lugar, se accedió a través de la pantalla inicial a la configuración de tipos de datos, ya que, antes de realizar el diagrama se deben crear los tipos de datos para luego asignárselos a las características. Se crearon los siguientes tipos de datos:

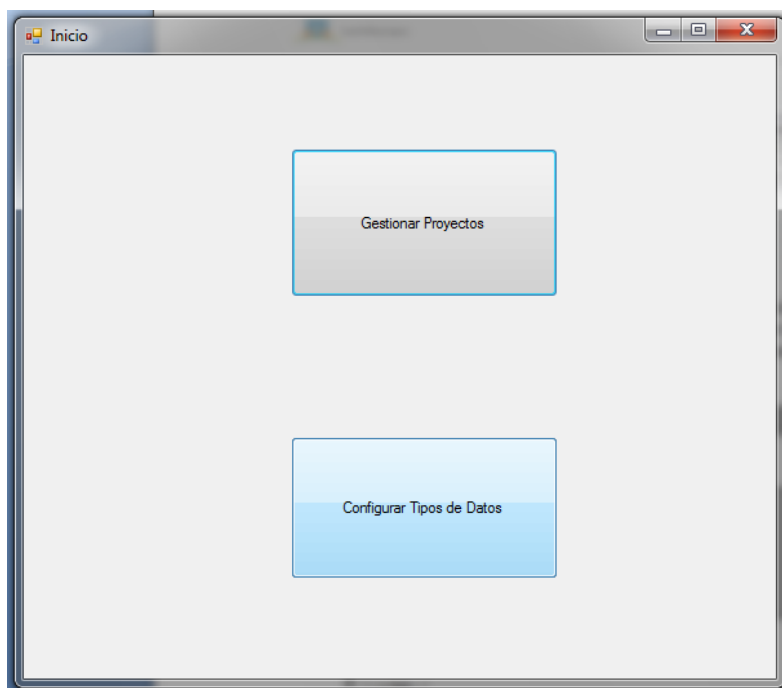


Ilustración 38 - Pantalla de inicio

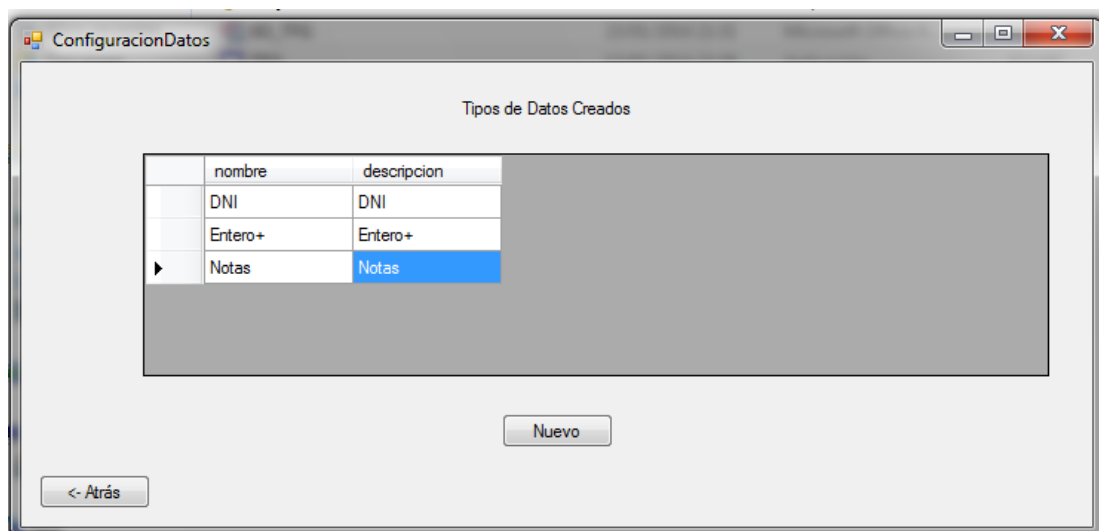
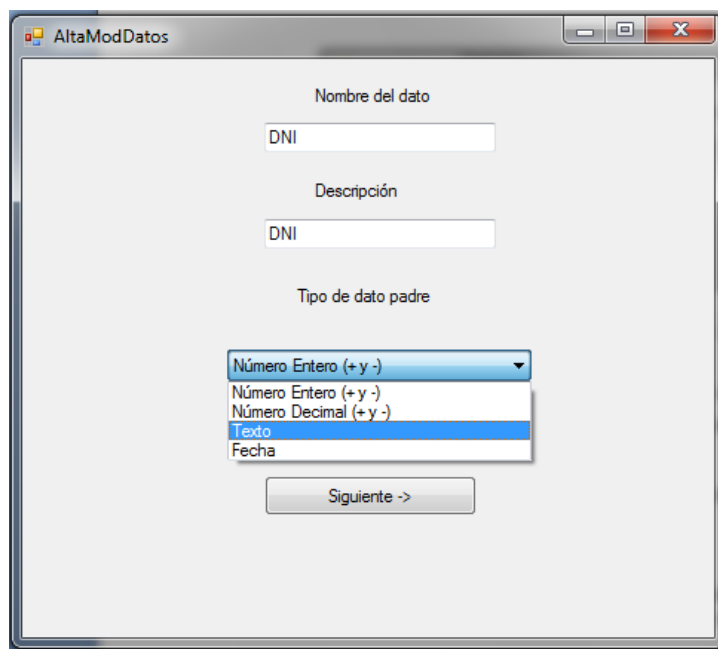


Ilustración 39 - Pantalla Gestión tipos de datos

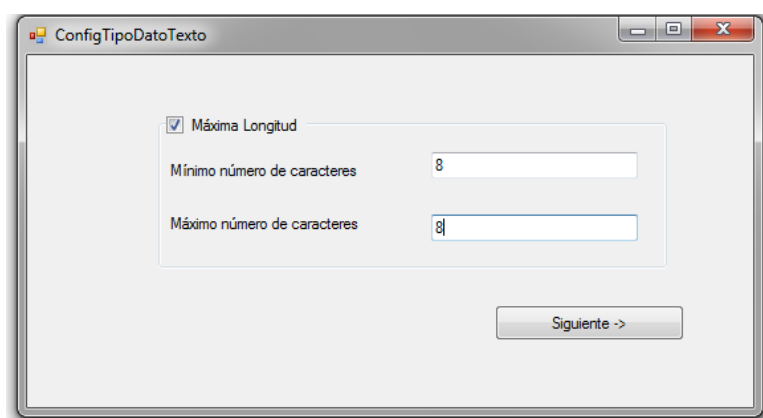
Los tipos de datos tienen la siguiente configuración:

- DNI: para gestionar los DNIs de los alumnos mediante cadenas de caracteres de 8 dígitos. Es un tipo de dato que hereda del tipo de dato primitivo “Texto”.



The screenshot shows a window titled 'AltaModDatos'. It contains three text input fields: 'Nombre del dato' with the value 'DNI', 'Descripción' with the value 'DNI', and 'Tipo de dato padre'. Below these fields is a dropdown menu for 'Tipo de dato padre' with the following options: 'Número Entero (+ y -)', 'Número Entero (+ y -)', 'Número Decimal (+ y -)', 'Texto' (which is selected), and 'Fecha'. At the bottom of the window is a button labeled 'Siguiente ->'.

Ilustración 40- Alta tipo de dato DNI 1



The screenshot shows a window titled 'ConfigTipoDatoTexto'. It contains a checkbox labeled 'Máxima Longitud' which is checked. Below this checkbox are two text input fields: 'Mínimo número de caracteres' with the value '8' and 'Máximo número de caracteres' with the value '8'. At the bottom of the window is a button labeled 'Siguiente ->'.

Ilustración 41 - Alta tipo de dato DNI 2

- Calificación: para asignárselo a los atributos “nota1” y “nota2”. Son valores decimales que van del 1 al 10. Tienen la siguiente configuración:

The screenshot shows a window titled 'AltaModDatos'. It has three text input fields: 'Nombre del dato' (containing 'Calificación'), 'Descripción' (containing 'Calificación'), and 'Tipo de dato padre'. Below these is a dropdown menu for 'Tipo de dato padre' with the following options: 'Número Decimal (+ y -)', 'Número Entero (+ y -)', 'Número Decimal (+ y -)' (highlighted), 'Texto', and 'Fecha'. At the bottom is a 'Siguiente ->' button.

Ilustración 42 - Alta tipo de dato Calificación 1

The screenshot shows a window titled 'ConfigTipoDatoNumero'. It has several sections:

- Entre:** A checked checkbox. Below it are 'Desde' (1) and 'Hasta' (10) input fields.
- Decimales:** A checked checkbox. Below it is 'Máximo Número' (2) and radio buttons for 'Punto' and 'Coma' (selected).
- Porcentaje:** An unchecked checkbox.
- Separador de miles:** An unchecked checkbox with radio buttons for 'Punto' and 'Coma'.
- Moneda:** An unchecked checkbox with radio buttons for '€', '\$', and '£', and an 'Otro' option with an input field.

 At the bottom right is a 'Siguiente ->' button.

Ilustración 43 - Alta tipo de dato Calificación 2

El tipo de dato de calificación es un número decimal del 1 al 10, con dos números máximos de decimales y con una coma como separador.

- Nota: es un dato de tipo “Enum”, es decir, un dato con una lista de valores determinados. Puesto que estos valores son de tipo Texto, su configuración es la siguiente:

Ilustración 44 - Alta tipo dato Nota 1

Ilustración 45 - Alta tipo dato Nota 2

Una vez creados los tipos de datos, se accedió a la pantalla de proyectos, para crear un nuevo proyecto:

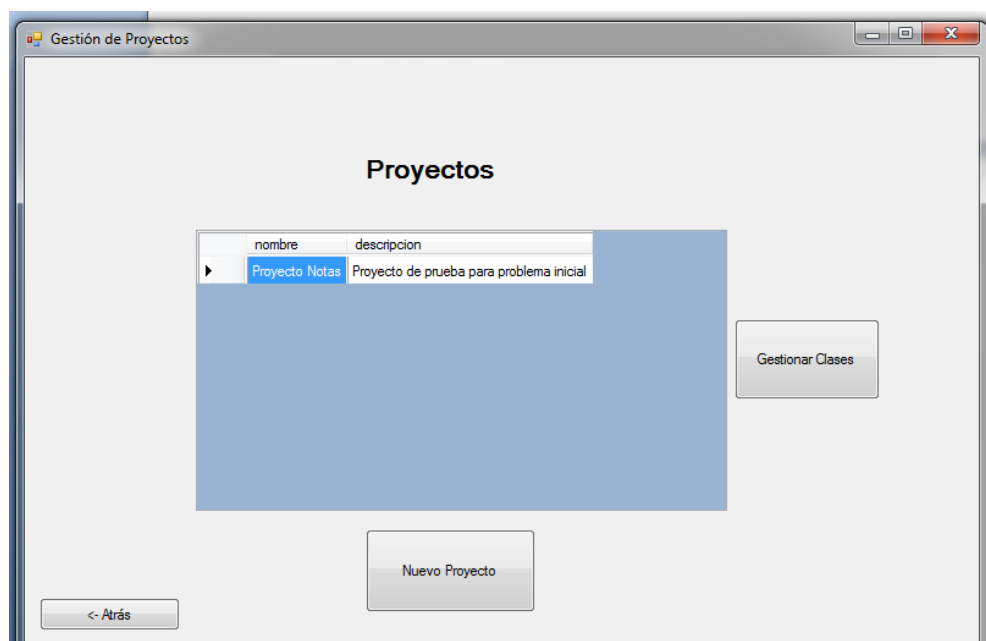


Ilustración 46 - Pantalla gestión de proyectos

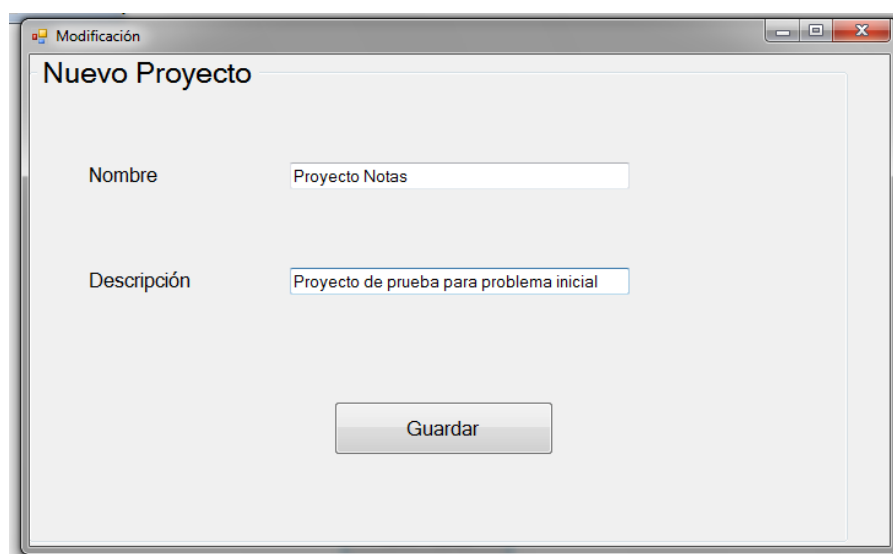
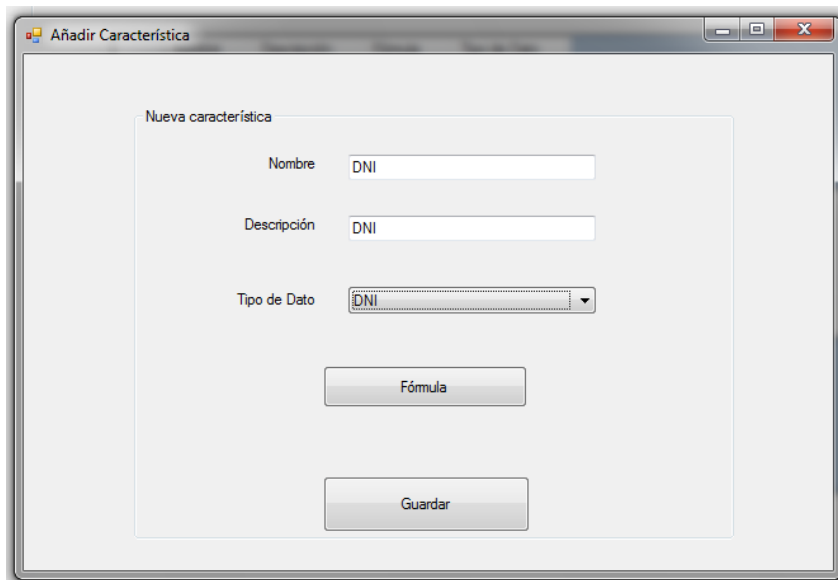


Ilustración 47 - Pantalla nuevo proyecto

A continuación se comenzó la gestión de sus clases, comenzando por la de “alumno”.

Puesto que alumno no contiene objetos de otra clase, se configuraron únicamente sus características:



Añadir Característica

Nueva característica

Nombre: DNI

Descripción: DNI

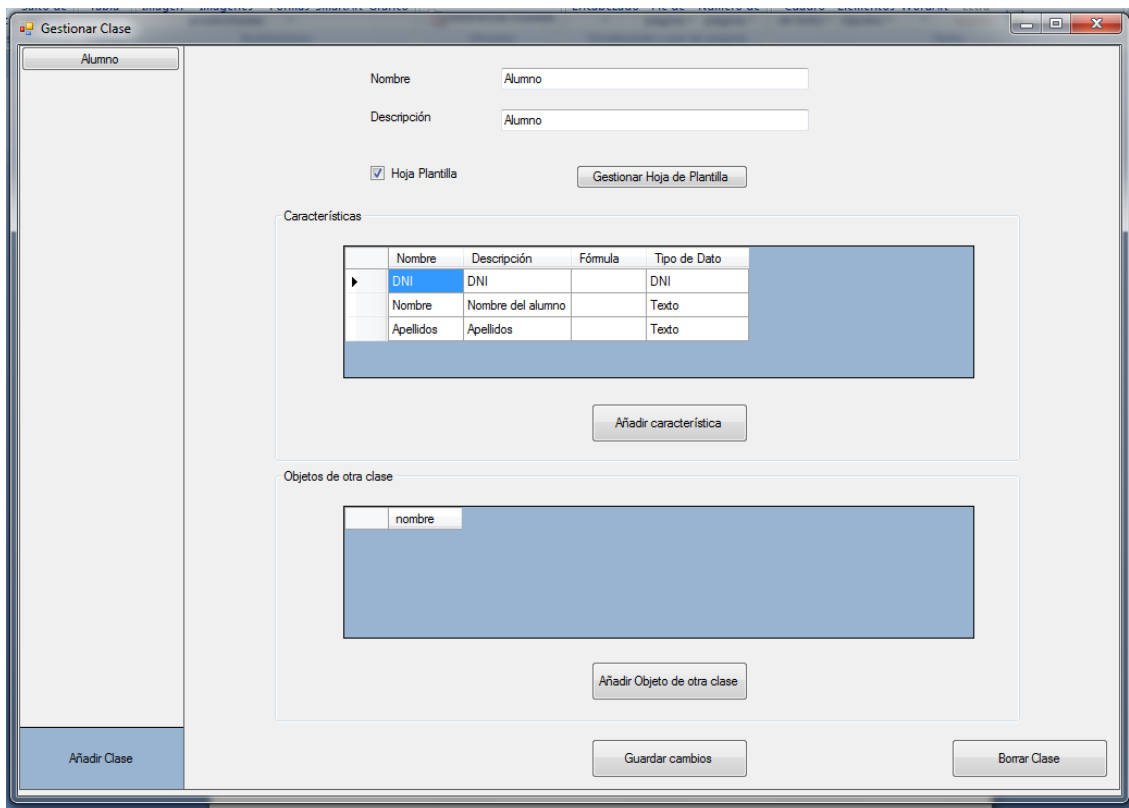
Tipo de Dato: DNI

Fórmula

Guardar

Ilustración 48 - Alta características alumno

El resultado final fue el siguiente:



Gestionar Clase

Alumno

Nombre: Alumno

Descripción: Alumno

☒ Hoja Plantilla Gestionar Hoja de Plantilla

Características

	Nombre	Descripción	Fórmula	Tipo de Dato
▶	DNI	DNI		DNI
	Nombre	Nombre del alumno		Texto
	Apellidos	Apellidos		Texto

Añadir característica

Objetos de otra clase

nombre

Añadir Objeto de otra clase

Añadir Clase Guardar cambios Borrar Clase

Ilustración 49 - Configuración alumno final

La hoja de plantilla correspondiente se configuró como estática, ya que, el resto de tipos de plantilla no tienen sentido en un caso como este:

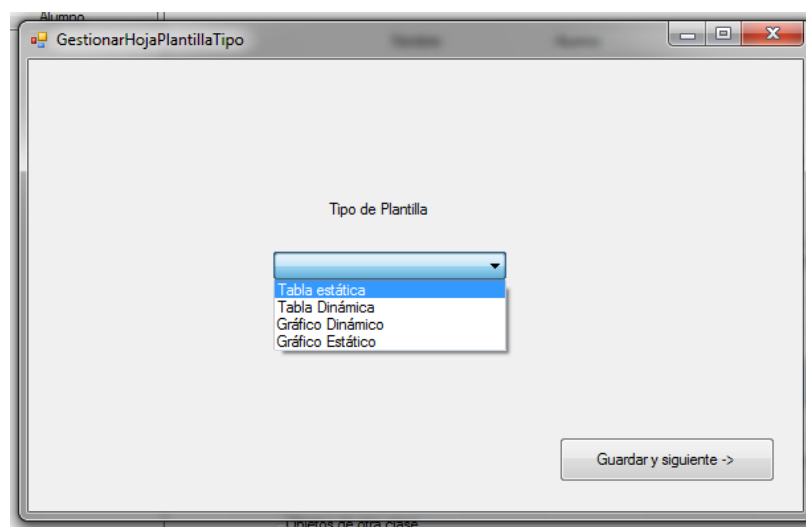


Ilustración 50 - Configuración Hoja Plantilla Alumno

A continuación, se configuraron las propiedades de los atributos del alumno en la hoja de plantilla:

- Debido a que el DNI es único por alumno, no debe permitir valores repetidos y debe ser obligatoria.
- El nombre y los apellidos, deben ser obligatorios pero pueden tener valores repetidos.

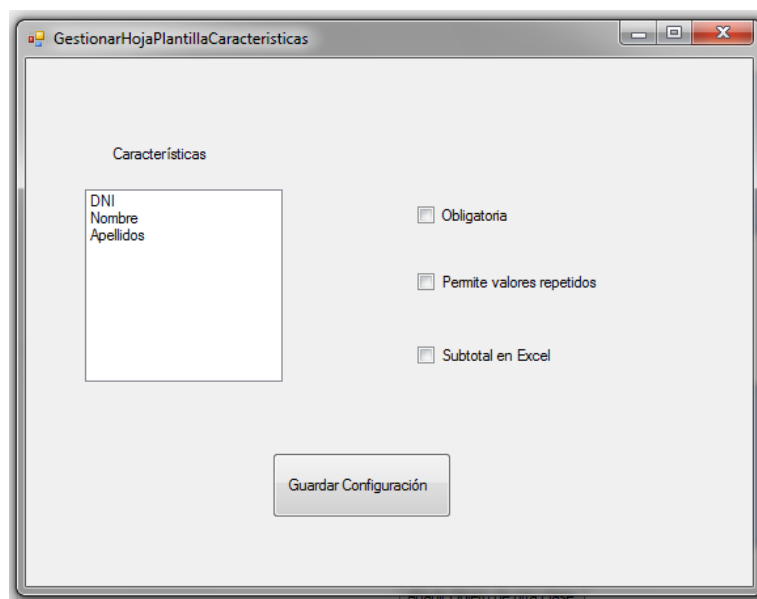


Ilustración 51 - Configuración Características Hoja Plantilla Alumno

El paso siguiente fue cargar la hoja. Primeramente se hizo desde la aplicación:

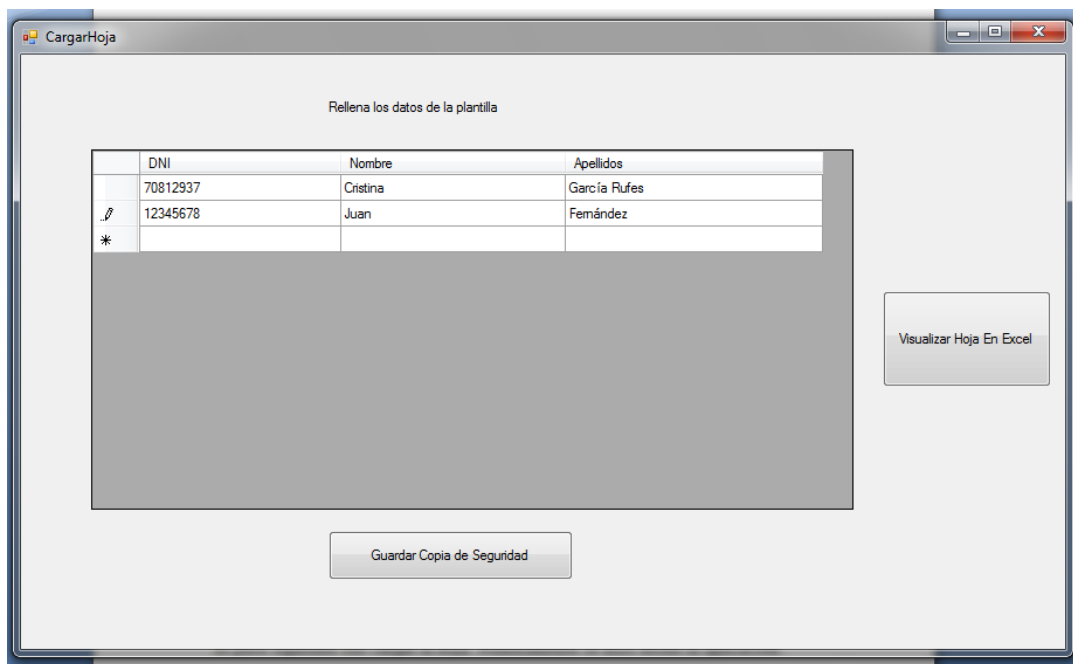


Ilustración 52 - Carga de datos Alumno

Para probar que se había generado la tabla acorde a los requisitos especificados, se probó a introducir valores incorrectos:

- Longitud del DNI > 8 caracteres:

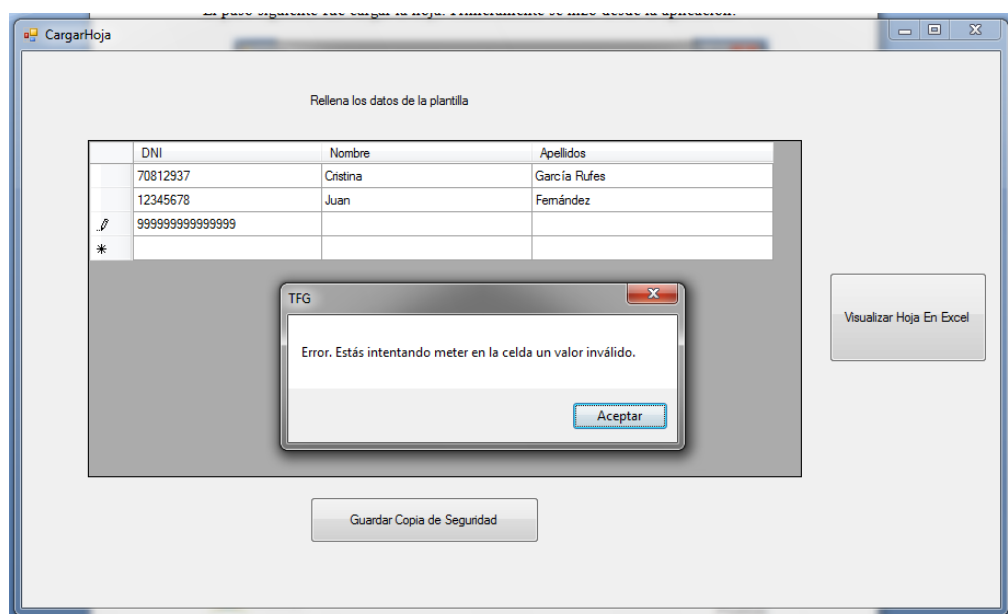


Ilustración 53 - Prueba Dato Incorrecto 1

- Longitud del DNI < 8 caracteres:

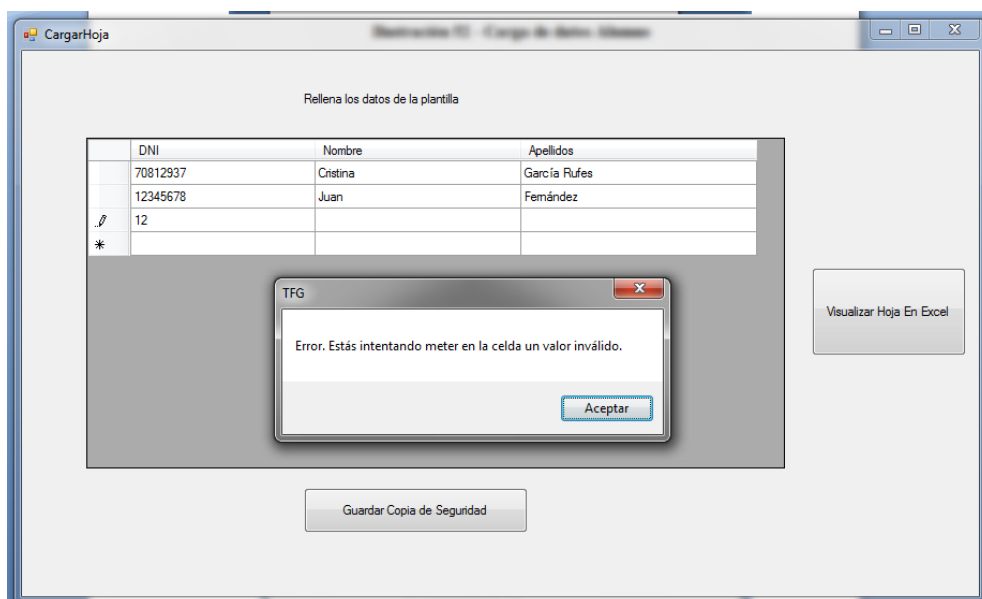


Ilustración 54 - Prueba Dato Incorrecto 2

- DNI repetido:

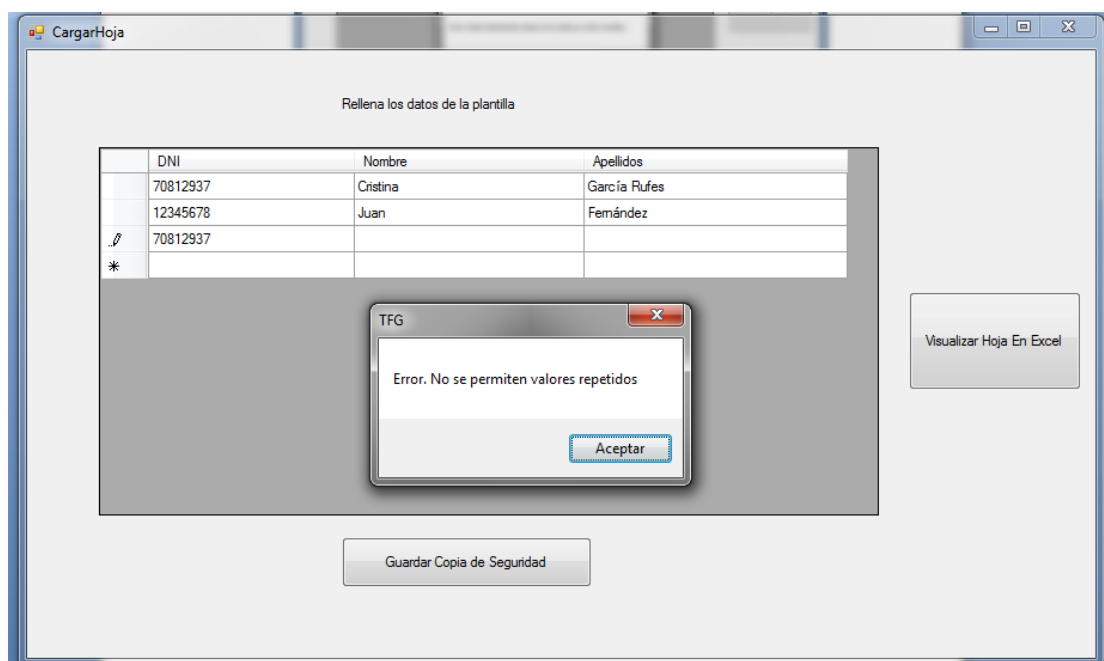


Ilustración 55 - Prueba Dato Incorrecto 3

- DNI en blanco:

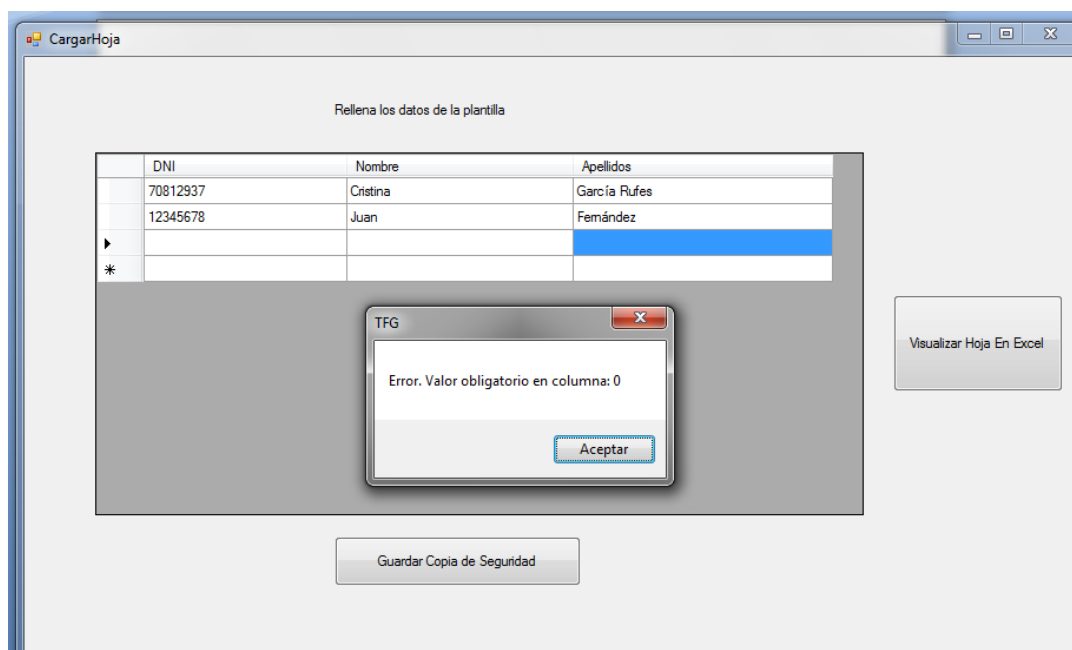


Ilustración 56 - Prueba Dato Incorrecto 4

Como se puede observar, la aplicación mostraba un mensaje de error en caso de introducir valores que no coincidan con la configuración especificada.

- Nombre repetido: Resultado correcto

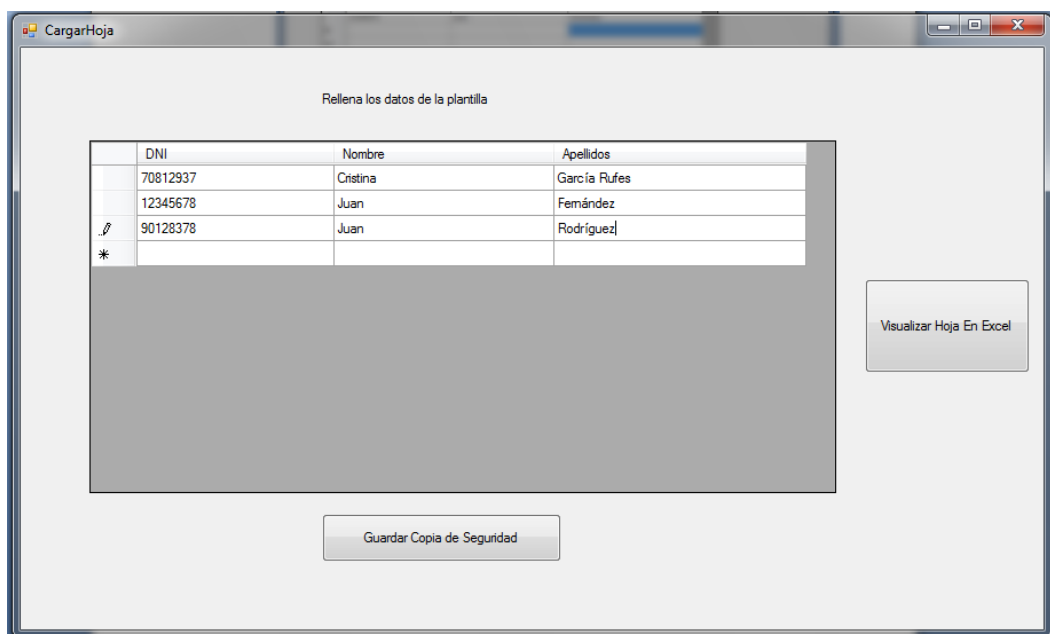


Ilustración 57 - Inserción dato correcto

La siguiente comprobación era ver si, en Excel, la generación de la hoja de cálculo se hacía cumpliendo con los mismos requisitos.

La hoja que se generó fue en forma de tabla estática con un botón al lado que permitía guardar en la copia de seguridad, incluyendo filtros para una mejor visualización de los datos:

	A	B	C	D	E	F	G	H
1	Alumno							
2	DNI	Nombre	Apellidos					
3	70812937	Cristina	García Rufes					
4	12345678	Juan	Fernández					
5	90128378	Juan	Rodríguez					
6								
7								
8								
9								
10								

Guarda copia de seguridad

Ilustración 58 - Hoja de cálculo generada

Al realizar las mismas pruebas, se encontraron los siguientes resultados:

- DNI incorrecto:

	A	B	C	D	E	F	G	H	I	J
1	Alumno									
2	DNI	Nombre	Apellidos							
3	70812937	Cristina	García Rufes							
4	12345678	Juan	Fernández							
5	90128378	Juan	Rodríguez							
6	999999999									
7										
8										
9										
10										
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										

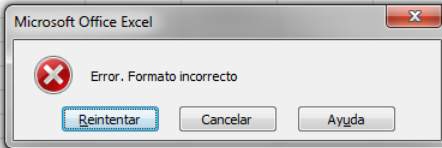


Ilustración 59 - Inserción dato incorrecto hoja de cálculo

- DNI duplicado:

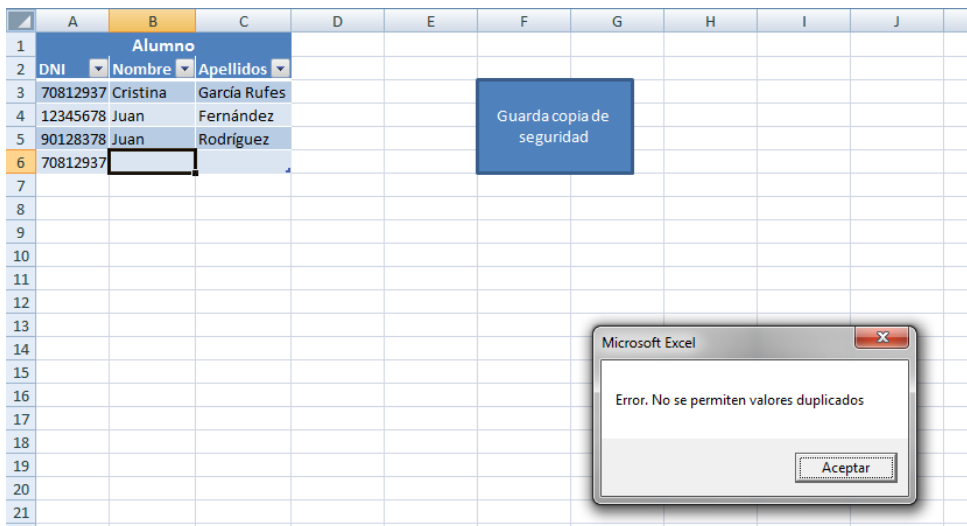


Ilustración 60 - Inserción dato duplicado hoja de cálculo

- DNI blanco:

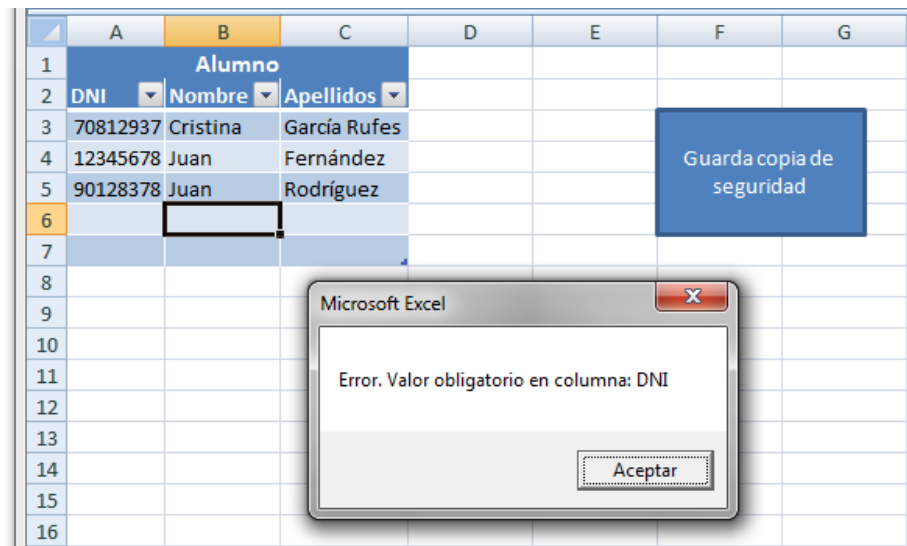


Ilustración 61 - Inserción errónea dato obligatorio hoja de cálculo

La siguiente prueba que se realizó fue tratar de guardar los datos de la hoja de plantilla en la copia de seguridad, obteniendo como resultado:

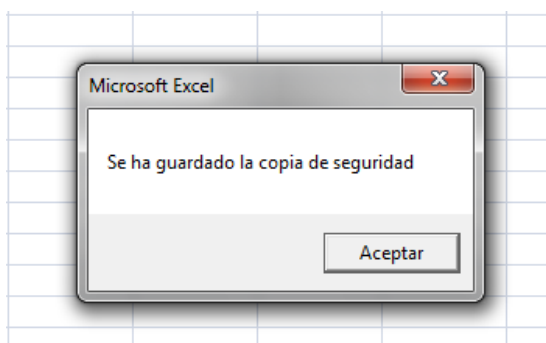


Ilustración 62 - Copia de seguridad almacenada

A continuación se configuró la clase Examen, en la cual se creó una característica con una fórmula, probando en primer lugar a insertar una fórmula que no era correcta ya que la variables de esa fórmula no existen:

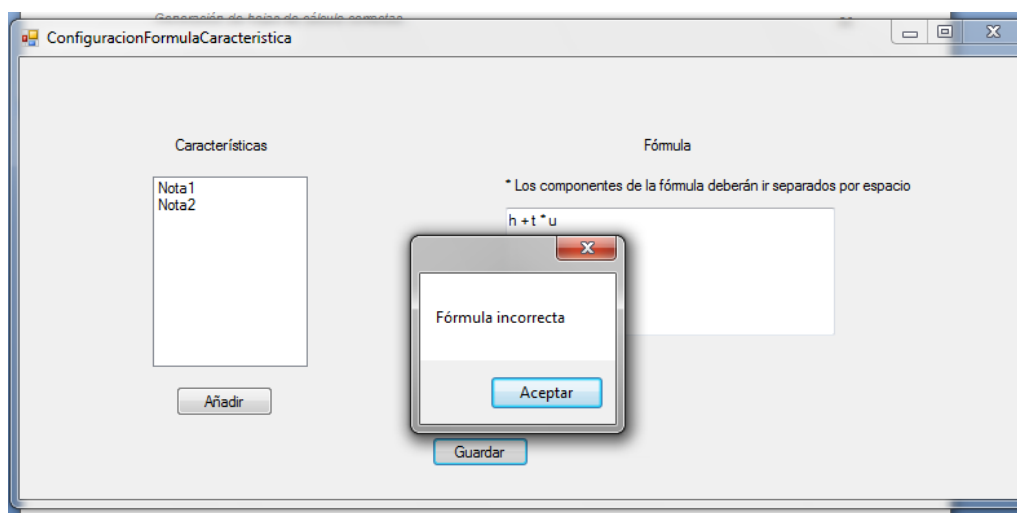


Ilustración 63 - Fórmula Incorrecta

Se añadió el objeto Alumno a la clase Examen y el resultado final de la clase Examen fue el siguiente:

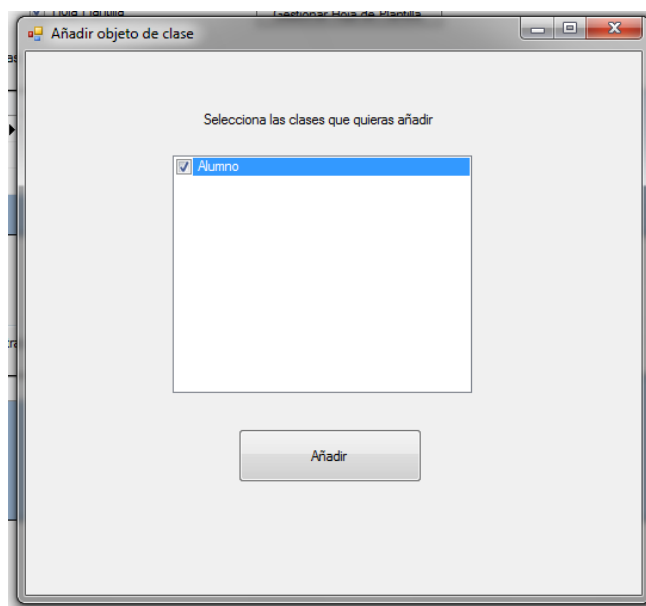


Ilustración 64 - Añadir Objeto a Otra clase

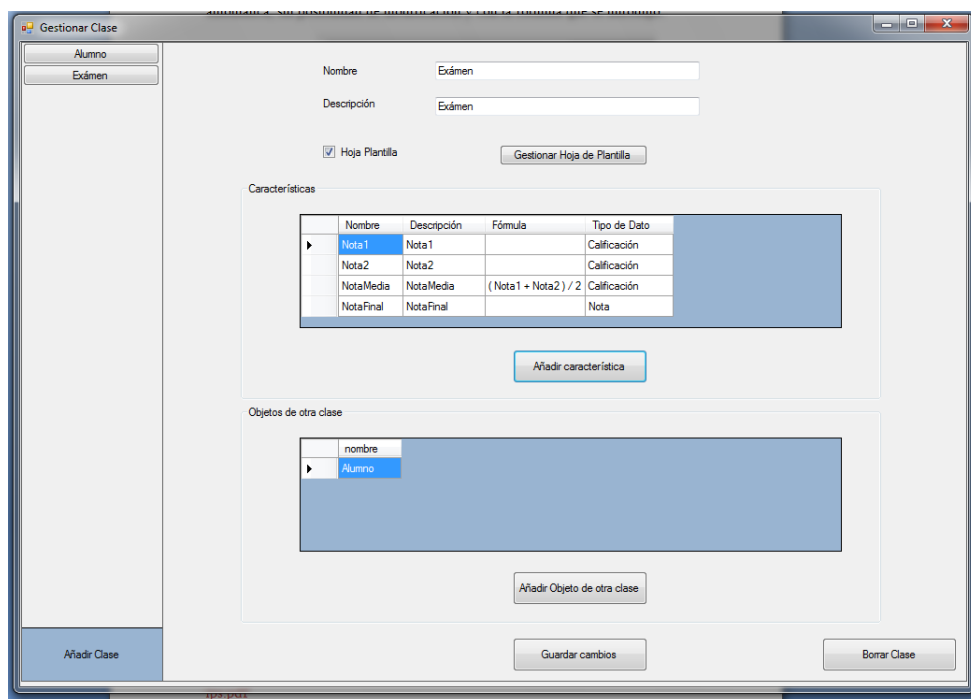


Ilustración 65 - Resultado final clase examen

Puesto que la clase Examen contiene objetos de Alumno, en la configuración de la hoja de plantilla se especificó qué atributos de Alumno se deseaban añadir en la hoja de plantilla. Se probó a tratar de añadir más de uno, algo que la aplicación no debe permitir debido al requisito funcional especificado en la Tabla 14 - Requisito RSF_14.

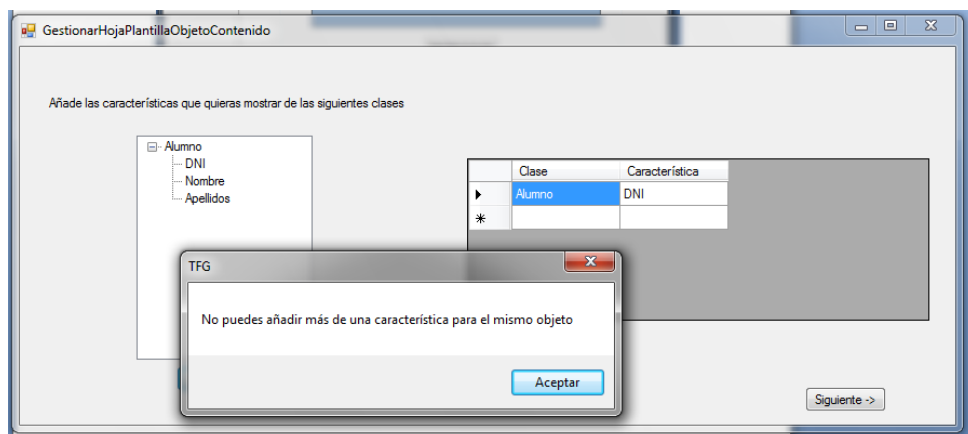


Ilustración 66- Error al añadir dos referencias al mismo objeto

La tabla resultante que se generó incluía el DNI del alumno, por lo que las pruebas que se hicieron, consistieron, en primer lugar, en introducir un alumno que no existía:

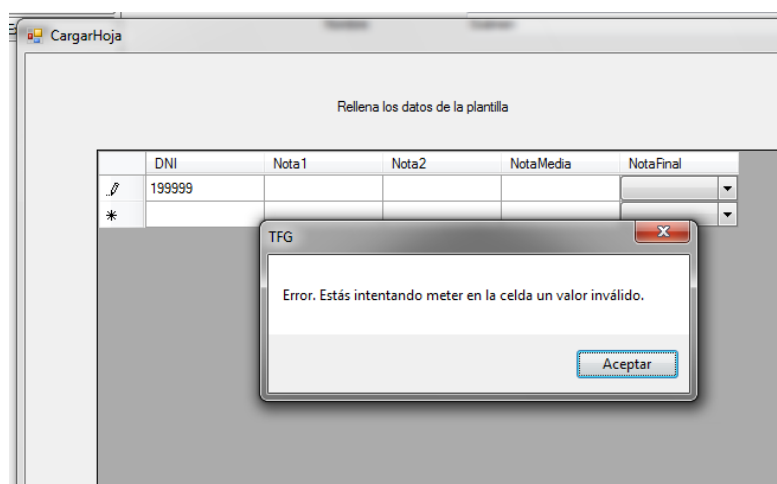


Ilustración 67 - Inserción referencia inexistente

Se comprobó también la funcionalidad de autocompletar con los datos del alumno:

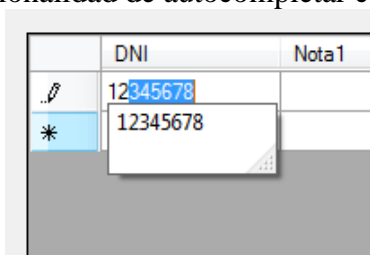


Ilustración 68 – Autocompletar

Puesto que el tipo de dato de “Calificación” se definió como un decimal entre 1 y 10, se probaron inserciones inválidas:

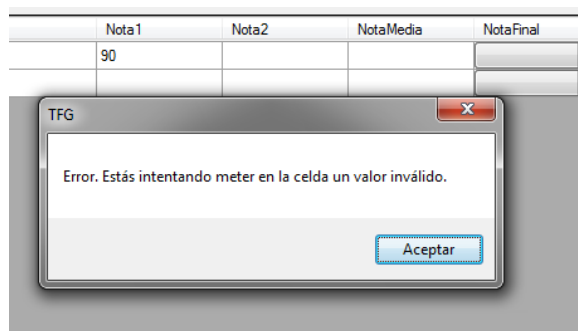


Ilustración 69 - Inserción nota inválida

Como se puede observar en la siguiente imagen, la nota media se calcula de forma automática, sin posibilidad de modificación y con la fórmula que se introdujo. Además se puede observar que el tipo enumerado es una lista de datos cerrada en la que sólo se pueden elegir los valores que se introdujeron en la configuración de los datos:

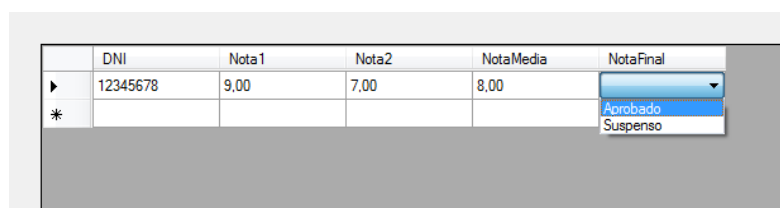


Ilustración 70 - Fórmulas y Datos Enum

En Excel se hicieron las mismas pruebas y los resultados obtenidos fueron los siguientes:

- Para la enumeración de datos, se observa que se ha creado una validación de datos con una lista desplegable, por lo que el usuario no puede introducir datos que no se encuentren en esta lista.

Exámen				
DNI	Nota1	Nota2	NotaMedia	NotaFinal
12345678	2	4	3	Suspense

Ilustración 71 - Validación de datos en lista Excel

- Fórmulas automáticas:

Exámen				
DNI	Nota1	Nota2	NotaMedia	NotaFinal
12345678	2	4	=([Nota1]+[Nota2])/2	

Ilustración 72 - Fórmulas automáticas

- Referencias a objetos de otra tabla:

Exámen				
DNI	Nota1	Nota2	NotaMedia	NotaFinal
99999999	3	4	3,5	Aprobado

Ilustración 73 - Referencias a objeto de otra tabla

Como se puede observar, la introducción y modificación de los datos está controlada por diferentes métodos de validación, tanto en la aplicación como en la propia hoja de cálculo. Esto quiere decir que los datos que se guarden en la copia de seguridad van a ser correctos.

Este ejemplo está orientado a los tipos de usuario que son programadores y que no tienen experiencia en el uso de las hojas de cálculo. Lo que está haciendo realmente la aplicación, es evitar que este tipo de usuario tenga que configurar los estilos de la hoja, las validaciones de los datos y las referencias entre celdas. Así se consigue que una persona con conocimientos en programación y capacidad de realizar un diagrama de clases de solución a un problema a través de una hoja de cálculo sin necesidad de entender el funcionamiento de Excel.

7.2. Gestor de Ventas

El siguiente caso de estudio está orientado a usuarios más expertos en Excel pero sin conocimientos en programación. Este tipo de usuario puede ser cualquier empresa que necesite generar informes que cumplan con unos requisitos concretos para el estudio de algunos resultados. Es muy común el uso de Excel en estos casos debido a su amplia gama de tipos de informe.

Como ejemplo se puede plantear el siguiente problema:

- *Se necesita generar un informe en una hoja de cálculo Excel en el que se visualicen de forma sencilla la venta de tres productos concretos que vende la empresa, indicando su fecha de incorporación, el precio de venta, la cantidad vendida y el país donde se vendió. La lista de los países se obtendrá de una tabla país que proviene de otra hoja de cálculo. Será necesario que las columnas tengan un formato correcto ya que este informe se tomará como base para el estudio de las ventas totales por país.*

Un usuario con algo de experiencia en Excel y que haya utilizado tablas dinámicas alguna vez, pensará como primera solución en construir el informe a través de una de estas tablas, debido a su sencillez a la hora de gestionar los datos. Éste tipo de usuario, sabe además, que estas tablas toman como referencia una tabla estática en la que se introducen los datos. Sin embargo, no conoce el uso de los diagramas de clase y no tiene experiencia en programación. Esta aplicación se puede adaptar a estos usuarios de forma muy simple ya que se puede entender una “clase” como una “hoja”, donde las “características” de la clase equivalen a las “columnas”. Con un pequeño manual de usuario y alguna explicación conceptual puede llegar a manejar la aplicación, tanto para la gestión de clases como para los tipos de datos, ya que normalmente, los usuarios que han usado Excel conocen lo que significan los tipos de datos de las columnas.

La solución a este problema se podría dar de la siguiente forma:

- En primer lugar, se crearía una tabla país, que contendría el nombre del mismo (en formato texto).
- La carga de esta tabla se podría hacer, o bien desde la propia aplicación o bien desde Excel. En cualquier caso, no sería necesario cargar los datos uno a uno ya que en ambas opciones se permite el uso de “copiar y pegar”. Cuando se copian y pegan los datos, también se están ejecutando las validaciones, ya que éstas funcionan en tiempo real, por lo que esta acción no permitiría introducir datos inválidos.

Un posible ejemplo de la construcción de la clase país sería el siguiente:

Nombre	Descripción	Fórmula	Tipo de Dato
Nombre País	Nombre País		Texto

Ilustración 74 – Configuración Clase País

Nombre País
Canadá
México
EEUU
Francia
Alemania

Ilustración 75 - Tabla País Aplicación

	A	B
1	País	
2	Nombre País	
3	Canadá	
4	México	
5	EEUU	
6	Francia	
7	Alemania	
8		

Ilustración 76 - Tabla País Excel

- A continuación se puede crear una clase para el propio informe. Este informe está compuesto por:
 - Producto
 - Puesto que son 3 productos en concreto se podría crear un tipo de dato “Producto” o bien crear una tabla producto y luego referenciarla. En ambos casos, no se permitiría introducir productos que no estén o en el tipo de dato, o en la tabla Producto.
 - Fecha de incorporación:
 - Ya que es un informe para una presentación y estudio de resultados, debe tener un formato propio de fechas.
 - Precio de venta:
 - En formato de moneda, y con separador tanto de decimales como de miles, para que el informe se visualice mejor.
 - Cantidad Vendida:
 - Número entero > 0
 - Se podría crear un total que fuera el resultado de (precio de venta * cantidad vendida) para el producto en concreto, en la fecha elegida y en el país al que se refiere.
 - País
 - Obtenido de la tabla país.

Así pues, la clase informe quedaría configurada de la siguiente manera:

Gestionar Clase

País: **Ventas Totales**

Nombre: Ventas Totales

Descripción: Ventas Totales

☒ Hoja Plantilla Gestionar Hoja de Plantilla

Características

	Nombre	Descripción	Fórmula	Tipo de Dato
▶	Producto	Producto		Productos
	Fecha Incorporación	Fecha Incorporación		Fecha
	Precio/Venta	Precio/Venta		Dinero/Ventas
	Cantidad/Vendida	Cantidad/Vendida		Entero+
	Total	Total	Cantidad/Vendida * Precio/Venta	Entero+

Añadir característica

Objetos de otra clase

	nombre
▶	País

Añadir Objeto de otra clase

Añadir Clase Guardar cambios Borrar Clase

Ilustración 77 - Clase informe de ventas

Y los tipos de datos de la siguiente forma:

- DineroVentas:

Ilustración 78 - Tipo de dato Dinero Ventas

- Producto:

Ilustración 79 - Tipo dato Producto

Lo más fácil para una correcta visualización de los datos, como se ha dicho anteriormente, es una tabla dinámica. La carga de esta tabla se puede realizar a través de la aplicación y generar en Excel una hoja dinámica con los datos introducidos:

- Un ejemplo de carga puede ser el siguiente:

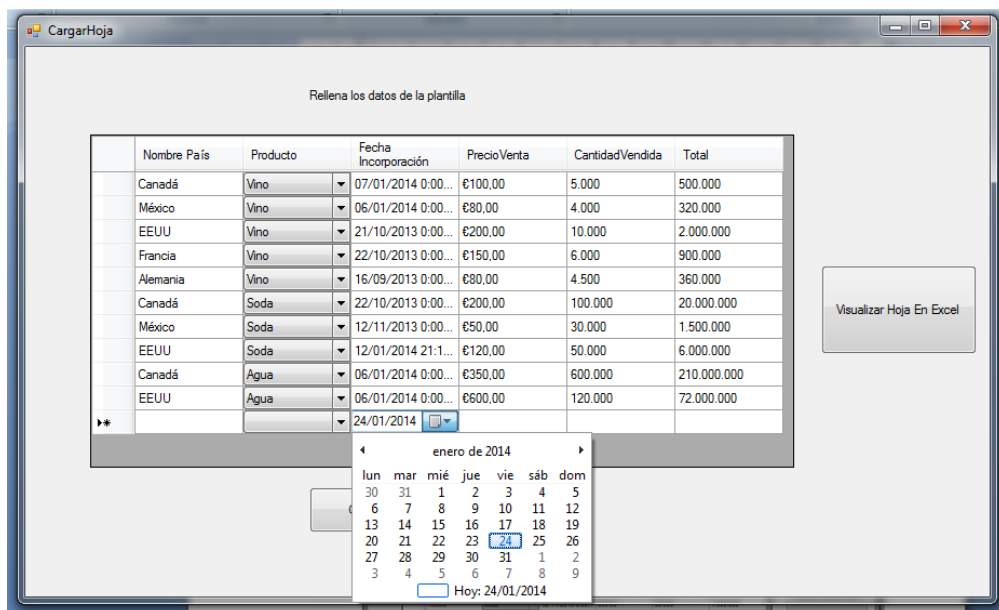


Ilustración 80 - Carga del informe de ventas

Como se puede observar, las columnas tienen un formato que a simple vista resulta atractivo, ya que los números tienen separador de miles, el precio está en euros y la fecha se coge directamente del calendario, sin posibilidad de introducir datos incorrectos.

La configuración de la hoja dinámica queda de la siguiente forma:

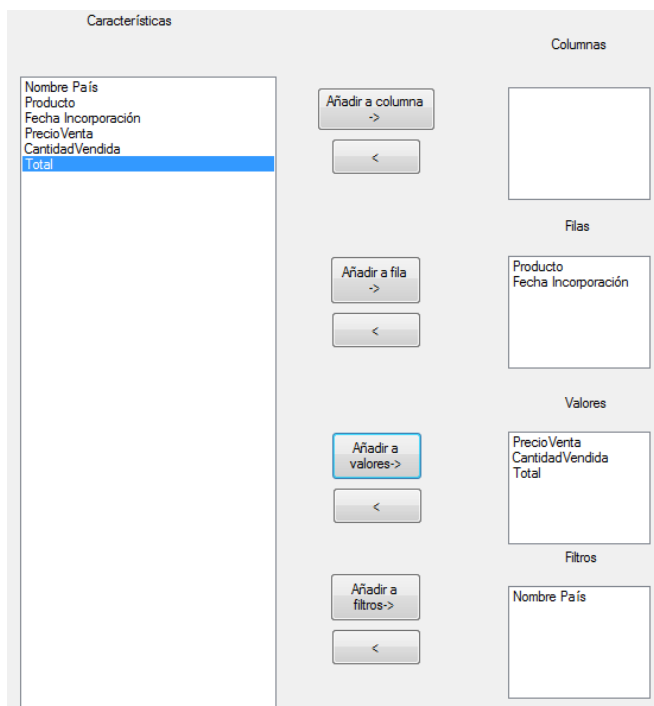


Ilustración 81 - Informe dinámico ventas

Con esta configuración se generaría la siguiente hoja de Excel:

Valores			
Rótulos de fila	Suma de PrecioVenta	Suma de CantidadVendida	Suma de Total
Agua	950 €	720.000	282.000.000
01/06/2014	950 €	720.000	282.000.000
Soda	370 €	180.000	27.500.000
22/10/2013 0:00:00	200 €	100.000	20.000.000
11/12/2013	50 €	30.000	1.500.000
01/12/2014 21:16	120 €	50.000	6.000.000
Vino	610 €	29.500	4.080.000
16/09/2013 0:00:00	80 €	4.500	360.000
21/10/2013 0:00:00	200 €	10.000	2.000.000
22/10/2013 0:00:00	150 €	6.000	900.000
01/06/2014	80 €	4.000	320.000
01/07/2014	100 €	5.000	500.000
(en blanco)			
24/01/2014 2:09			
Total general	1.930 €	929.500	313.580.000

Ilustración 82 - Informe final País

Como se puede observar, este informe es un informe completo que da solución al problema propuesto, ya que se pueden visualizar las ventas totales por país, por producto, por fecha, etc y que aseguran que los datos están bien introducidos. Asegura que no se ha introducido ningún producto incorrecto, que el precio y la cantidad vendida no son números negativos y que los países son los que estaban en la hoja de Excel.

7.3. Analizador Financiero

Un usuario que sea especialista en Excel utilizará la hoja de cálculo para realizar modelos y estudios financieros. Una forma de que evite problemas a la hora de introducir los datos es creando el modelo a través de la herramienta. Un usuario experimentado conoce diagramas de relación de ciertos elementos, referencias en las hojas de cálculo, fórmulas, tipos de datos, etc por lo que no tendría problemas en utilizar la aplicación.

El siguiente ejemplo muestra la generación de un informe de resultados de ratios financieros. Un ratio financiero es un coeficiente que proporcionan unidades contables y financieras para medir y analizar el estado actual de una organización. En este ejemplo se estudian la liquidez y el endeudamiento. El modelo de datos se crea de manera muy simple pero trata de demostrar el uso de la herramienta para generación de hojas de cálculo correctas. Este modelo se puede ampliar mucho más y se pueden utilizar fórmulas complejas, ya que la herramienta lo permite, pero en este caso se usará un ejemplo sencillo:

Se supone que se quiere generar un informe que estudie el los ratios de de endeudamiento y liquidez en el año 2010 y 2011 para todos los meses del año.

Se distinguen 3 clases:

- Años: podría ser un enumerado igual que en el caso de “producto”, pero en este caso se ha elegido como otra tabla aparte. Clase que contiene una característica de tipo texto:

	Año
▶	2011
	2010
*	

Ilustración 83 - Clase Años

- RatioFinanciero: indica el tipo de ratio financiero. Tendrá una característica donde se indicará el tipo en formato texto, pero como posible ampliación:
 - Se podría introducir la fórmula para la liquidez como (activo corriente/pasivo corriente) en caso de querer ampliar este analizador.
 - De igual forma, la razón de endeudamiento se puede definir como (pasivo circulante + pasivo a largo plazo)/total activos

	Tipo
▶	Liquidez
	Endeudamiento
*	

Ilustración 84 - Clase Ratio Financiero

- Meses: clase con una característica tipo texto que contiene el nombre del mes. Se ha elegido clase en vez de tipo de dato enumerado ya que se podría querer añadir más características como por ejemplo el mes en formato número.

	Meses
▶	Enero
	Febrero
	Marzo
	Abril
	Mayo
	Junio
	Julio
	Agosto
	Septiembre
	Octubre
	Noviembre
	Diciembre
*	

Ilustración 85 - Clase Meses

- InformeFinal: contendrá un subtipo ligado al tipo de ratio y su valor. Además tendrá referencias a las clases: años, meses y ratio financiero. El valor se ha escogido en este primer caso de estudio como Decimal+:

Características

Nombre	Descripción	Fórmula	Tipo de Dato
Subtipo	Subtipo		Texto
Valor	Valor		Decimal+

Añadir característica

Objetos de otra clase

nombre
Años
Ratio Financiero
Meses

Ilustración 86 - Clase InformeFinal

- Eligiendo el tipo de plantilla como estática se crearía el siguientes informe:

InformeFinal					
Año	Tipo	Subtipo	Valor	Meses	
2011	Liquidez	Liquidez Corriente	1,23	Enero	
2011	Liquidez	Liquidez Corriente	1,22	Febrero	
2011	Liquidez	Liquidez Corriente	1,22	Marzo	
2011	Liquidez	Liquidez Corriente	1,22	Abril	
2011	Liquidez	Liquidez Corriente	1,2	Mayo	
2011	Liquidez	Liquidez Corriente	1,18	Junio	
2011	Liquidez	Liquidez Corriente	1,18	Julio	
2011	Liquidez	Liquidez Corriente	1,19	Agosto	
2011	Liquidez	Liquidez Corriente	1,2	Septiembre	
2011	Liquidez	Liquidez Corriente	1,22	Octubre	
2011	Liquidez	Liquidez Corriente	1,21	Noviembre	
2011	Liquidez	Liquidez Corriente	1,23	Diciembre	
2011	Liquidez	Prueba Ácida	0,93	Enero	
2011	Liquidez	Prueba Ácida	0,87	Febrero	
2011	Liquidez	Prueba Ácida	0,85	Marzo	
2011	Liquidez	Prueba Ácida	0,86	Abril	
2011	Liquidez	Prueba Ácida	0,87	Mayo	
2011	Liquidez	Prueba Ácida	0,81	Junio	
2011	Liquidez	Prueba Ácida	0,81	Julio	
2011	Liquidez	Prueba Ácida	0,87	Agosto	
2011	Liquidez	Prueba Ácida	0,93	Septiembre	
2011	Liquidez	Prueba Ácida	0,94	Octubre	
2011	Liquidez	Prueba Ácida	0,92	Noviembre	
2011	Liquidez	Prueba Ácida	0,84	Diciembre	
2011	Endeudamie	Endeudamiento Patrimonial	1,26	Enero	
2011	Endeudamie	Endeudamiento Patrimonial	1,28	Febrero	
2011	Endeudamie	Endeudamiento Patrimonial	1,41	Marzo	
2011	Endeudamie	Endeudamiento Patrimonial	1,43	Abril	
2011	Endeudamie	Endeudamiento Patrimonial	1,57	Mayo	
2011	Endeudamie	Endeudamiento Patrimonial	1,58	Junio	
2011	Endeudamie	Endeudamiento Patrimonial	1,64	Julio	
2011	Endeudamie	Endeudamiento Patrimonial	1,51	Agosto	
2011	Endeudamie	Endeudamiento Patrimonial	1,42	Septiembre	
2011	Endeudamie	Endeudamiento Patrimonial	1,56	Octubre	
2011	Endeudamie	Endeudamiento Patrimonial	1,33	Noviembre	

Ilustración 87 - Ejemplo carga InformeFinal

- Otro informe característico podría ser en tipo dinámico:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	Año	(Todas)													
2															
3	Suma de Valor	Rótulos de columna													
4	Rótulos de fila	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Septiembre	Octubre	Noviembre	Diciembre	(en blanco)	Total general
5	Endeudamiento	4,22	4,26	4,32	4,36	4,54	4,56	4,59	4,4	4,33	4,54	4,22	4,42		52,76
6	Endeudamiento Patrimonial	3,22	3,26	3,32	3,36	3,54	3,56	3,58	3,42	3,34	3,52	3,26	3,4		40,78
7	Grado de propiedad	0,44	0,44	0,44	0,42	0,41	0,39	0,39	0,38	0,4	0,41	0,39	0,43		4,94
8	Solvencia	0,56	0,56	0,56	0,58	0,59	0,61	0,62	0,6	0,59	0,61	0,57	0,59		7,04
9	Liquidez	4,05	4,04	3,93	3,97	4,01	3,88	3,84	3,94	3,98	4,06	3,98	3,96		47,64
10	Liquidez Corriente	2,39	2,4	2,33	2,35	2,37	2,36	2,32	2,3	2,32	2,38	2,34	2,38		28,24
11	Prueba Ácida	1,66	1,64	1,6	1,62	1,64	1,52	1,52	1,64	1,66	1,68	1,64	1,58		19,4
12	(en blanco)														
13	(en blanco)														
14	Total general	8,27	8,3	8,25	8,33	8,55	8,44	8,43	8,34	8,31	8,6	8,2	8,38		100,4
15															
16															

Ilustración 88 - Ejemplo tabla dinámica InformeFinal

- E incluso un gráfico dinámico:

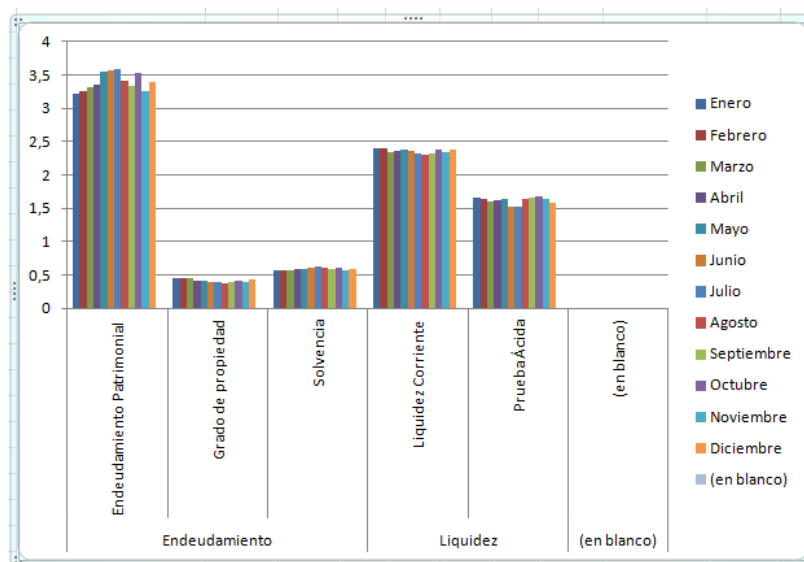


Ilustración 89 - Gráfico dinámico InformeFinal

La generación de estas hojas de cálculo es correcta debido a que las características se han definido como obligatorias y los valores deben ser decimales+. Además no se permite introducir tipos de ratio que no estén en la tabla.

Si se quisiera hacer un analizador se podría adaptar el tipo de dato “valor” a estudios más concretos. Si fuera más orientado al endeudamiento, por ejemplo, se podría definir valor como:

- Decimal entre 0,4 y 0,6 para valores óptimos.

De la misma forma se pueden crear fórmulas y más características relacionadas entre sí, controladas por la validación de los datos de la aplicación.

Se observa que todos los ejemplos están adaptados a diferentes tipos de usuario y generan hojas correctas, por lo que el objetivo de la aplicación queda cumplido.

8. Conclusiones y Futuras Líneas de Trabajo

En este punto se presentarán las conclusiones obtenidas realizando una comparación del objetivo final del proyecto con lo que se ha conseguido. Además se describirán los problemas encontrados en el desarrollo y la conclusión personal de este proyecto. Tras esto se propondrán algunas posibles mejoras y futuras líneas de trabajo.

8.1. Conclusión del trabajo realizado

Este proyecto tenía como objetivo implementar una herramienta de generación de hojas de cálculo correctas, tomando como referencia un diagrama de clases y de forma que pudiera utilizarse por diferentes tipos de usuarios. Se llevó a cabo un estudio de las funcionalidades de Excel para ver por qué era tan importante su uso y se analizaron sus diferentes tipos de usuario. Tras investigar las herramientas similares y sacar determinadas ideas de algunos tipos de tecnologías, se llevó a cabo la implementación de la herramienta. La aplicación debía cumplir con los siguientes requisitos principales:

- Permitir al usuario crear un diagrama de clases definiendo
 - Clases
 - Atributos para las clases
 - Tipos de datos asignados a los atributos

Como se ha podido ver en las pruebas de la aplicación, este requisito se ha cumplido de forma exitosa ya que el usuario puede crear clases, almacenarlas y relacionarlas entre sí.

- ✓ Permitir al usuario generar hojas de cálculo a partir del diagrama de clases configurado.

Este requisito también se ha cumplido con éxito ya que el usuario puede generar una hoja de plantilla que ha sido configurada en función de las características definidas para la clase. La hoja de cálculo generada cuenta con todos los sistemas de validación de datos para que el usuario no pueda introducir datos incorrectos.

- ✓ Adaptar la aplicación a diferentes tipos de usuario.

En la sección de pruebas se han estudiado tres casos diferentes en los que se comprobaba que la aplicación se puede utilizar en usuarios con distintos tipos de conocimientos, por lo que este requisito también se ha cumplido.

Se concluye, por tanto, que se ha alcanzado de manera exitosa el objetivo final de la aplicación.

8.2. Conclusión personal

A lo largo del desarrollo del proyecto he tenido dificultades en la implementación de algunas funcionalidades debido a las limitaciones de las hojas de cálculo y del lenguaje VBA. Aunque tengo experiencia laboral con este lenguaje, me han resultado complejas algunas partes del desarrollo por el código de bajo nivel utilizado y por ello he tenido que investigar y aprender nuevas tecnologías para poder cumplir con los requisitos.

Durante la carrera no he visto la plataforma .NET y aunque también tenía algo de experiencia laboral he tenido que formarme en este entorno, lo que en algún momento ha retrasado el desarrollo de la aplicación.

También he tenido que analizar las diferentes necesidades de los distintos tipos de usuario, lo que me ha ayudado a entender y comprender el trabajo de un ingeniero que se dedica al desarrollo de aplicaciones. Desde mi punto de vista es muy importante adaptar una solución propuesta al mayor número de gente posible, porque, si eso se consigue significa que el ingeniero ha conseguido de forma exitosa su objetivo. Gracias a la experiencia laboral adquirida con diferentes clientes he podido orientar mis ideas a las necesidades de los usuarios y por ello creo que los conocimientos adquiridos en la consultoría donde estoy trabajando han sido muy útiles.

Por último, hay que destacar que he podido poner en práctica todo lo aprendido en la universidad, desde la toma de requisitos, y pasando por la planificación, diseño e implementación hasta el desarrollo de la presente memoria.

Como conclusión personal puedo decir que este proyecto ha puesto a prueba mi capacidad como ingeniera y desde mi punto de vista, he superado la prueba. Estoy conforme con el trabajo realizado y me siento capacitada para enfrentarme a las diferentes adversidades que puedan surgir en el entorno laboral.

8.3. Posibles mejoras

A la hora de desarrollar una aplicación con un tiempo tan limitado es difícil saber y adaptar unas condiciones de satisfacción para que el trabajo se dé por concluido. En este caso la aplicación desarrollada contaba con una fecha de entrega cercana y no se podían llevar a cabo determinadas funcionalidades. Las posibles ampliaciones que se podrían realizar son las siguientes:

- En vez de tipos de datos generales que se utilicen para todos los proyectos, desglosar en tipos de datos generales y tipos de datos por proyecto.
- Crear perfiles de usuario:
 - Gestor: crea y configura las hojas de plantilla y las bloquea para evitar cambios.
 - Usuario: rellena las hojas de plantilla que el gestor le proporciona.
- Migrar a web para que el uso de la aplicación sea más numeroso. Las bases de datos de seguridad se pueden crear en un servidor y de esta forma crear plantillas que puedan usar varios usuarios a la vez.

La aplicación está preparada para posibles ampliaciones debido a su diseño de tres capas, por lo que no implicaría un problema de desarrollo.

8.4. Líneas futuras de trabajo

Esta aplicación puede utilizarse como sistema para trabajos futuros que realicen las siguientes funciones:

- Modelos financieros: ejemplos parecidos al analizador financiero explicado en el apartado de pruebas pero mucho más extenso, incluyendo características con fórmulas complejas, referencias a múltiples datos, etc.
- Generador de hojas de cálculo para problemas concretos que se definen con un diagrama de clases como solución tecnológica para aplicaciones de cliente.
 - Se puede evitar tener que implementar nuevas aplicaciones en Excel ya que la herramienta está preparada para generar las hojas con la configuración que se especifique.
 - Esto proporciona un abanico de posibilidades especialmente en clientes financieros.
- Motor de base de datos con referencias entre las tablas. Ya que se crean bases de datos de seguridad se puede plantear un futuro trabajo sacando más partido a esta funcionalidad.

9. Referencias

1. **Wikipedia.** "/Spreadsheet". <<http://en.wikipedia.org/wiki/Spreadsheet>>. [En línea] 2013.
2. **Cuesta, Yolanda.** "/Excel, la historia de las hojas de cálculo continua". <http://hojasdecalculo.about.com/od/Software_hojas_de_calculo/a/Excel-La-Historia-De-Las-Hojas-De-C-Alculo-Continua.htm>. [En línea]
3. **Stephen Bullen, Rob Bovey, John Green.** "*Professional Excel Development*".
4. **Lieberman, H., Paternò, F. and Wulf, V.** "*End User Development*". 2006.
5. **Worstell, Tim.** "/Microsoft's Excel Might Be The Most Dangerous Software On The Planet". <http://www.forbes.com/sites/timworstell/2013/02/13/microsofts-excel-might-be-the-most-dangerous-software-on-the-planet/>. [En línea]
6. **Kwak, James.** "/The Importance Of Excel". <<http://baselinescenario.com/2013/02/09/the-importance-of-excel/>>. [En línea]
7. **Monaghan, Peter.** "/They Said at First That They Hadn't Made a Spreadsheet Error, When They Had". <<http://chronicle.com/article/UMass-Graduate-Student-Talks/138763/>>. [En línea]
8. **Gregor Engels, Martin Erwig.** "/ClassSheets: Automatic Generation of Spreadsheet". <http://web.engr.oregonstate.edu/~erwig/papers/ClassSheets_ASE05.pdf>. [En línea] 2013.
9. **Protiviti.** <<http://www.protiviti.com/en-US/Documents/Resource-Guides/SpreadsheetRiskMgmtFAQ09.pdf>>. [En línea]
10. **Erwig, Martin.** "/Software Engineering for spreadsheets". <http://web.engr.oregonstate.edu/~erwig/papers/SEforSpreadsheets_IEEESoftware09.pdf>. [En línea]
11. **J'acome Cunha, Martin Erwig, João Saraiva.** "/Automatically Inferring ClassSheet Models from Spreadsheets". <http://web.engr.oregonstate.edu/~erwig/papers/InferCS_VLHCC10.pdf>. [En línea]
12. **Pool, Mark.** "/Ventajas y desventajas del uso de bases de datos en Access". <http://www.ehowenespanol.com/cuales-son-ventajas-desventajas-base-datos-microsoft-access-info_93194/>. [En línea] 2013.
13. **IEEE-STD-830-1998.** "*Especificaciones de los requisitos del software*".
14. **Andrew J KO, Robin Abraham, Laura Beckwith.** "/The State of the Art in End-User Software Engineering". <<http://faculty.washington.edu/ajko/papers/Ko2011EndUserSoftwareEngineering.pdf.pdf>>. [En línea]
15. **Kamalasen Rajalingham, David Chadwick, Brian Knight, Dilwyn Edwards.** "/Quality Control in Spreadsheets". <<http://panko.shidler.hawaii.edu/SSR/HICSS33/HICSS33-Rajal-SoftEngr.pdf>>. [En línea]
16. **Cunha, J'acome.** "/Towards a Catalog of Spreadsheet Smells?". <<http://alfa.di.uminho.pt/~jacome/down/iccsa12.pdf>>. [En línea]
17. **—.** "/Model-based Programming Environments for". <<http://alfa.di.uminho.pt/~jacome/down/sblp12.pdf>>. [En línea]

10. ANEXOS

ANEXO 1 – Imágenes diagrama de clases

Clases de la Capa_BD

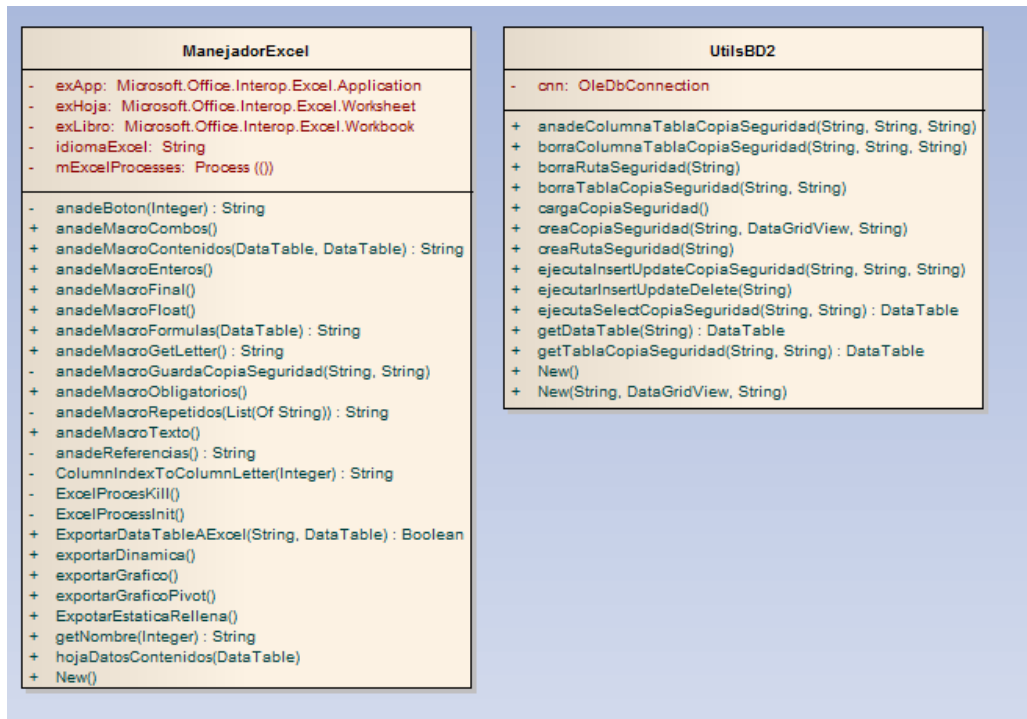


Ilustración 90 - Clases Capa BD

Clases Capa_Objeto_BD

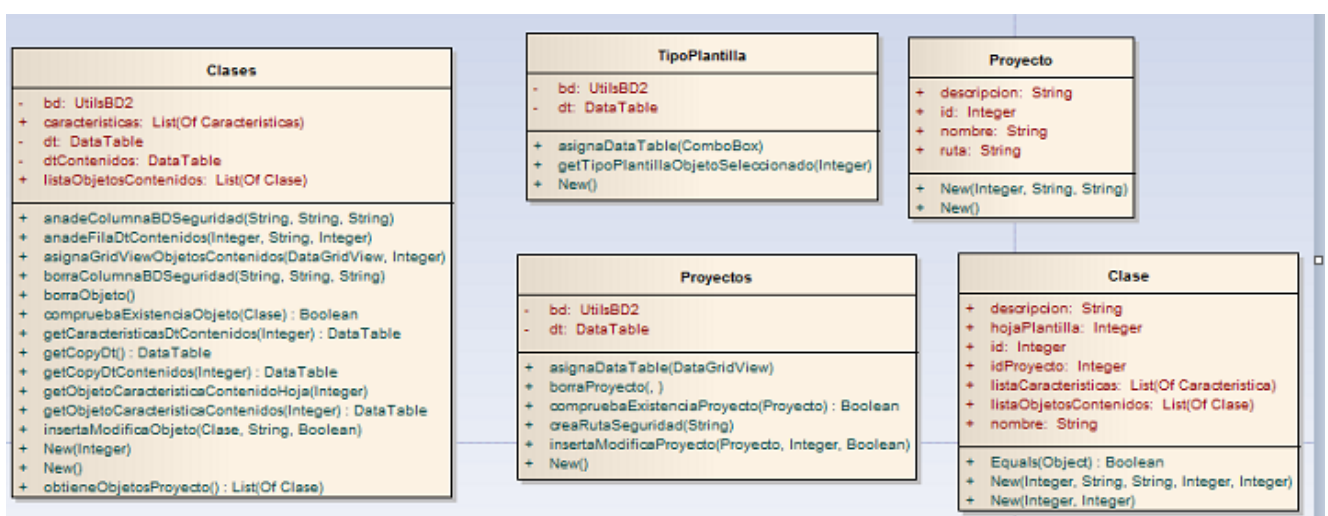


Ilustración 91 - Clases Capa_Objeto_BD1

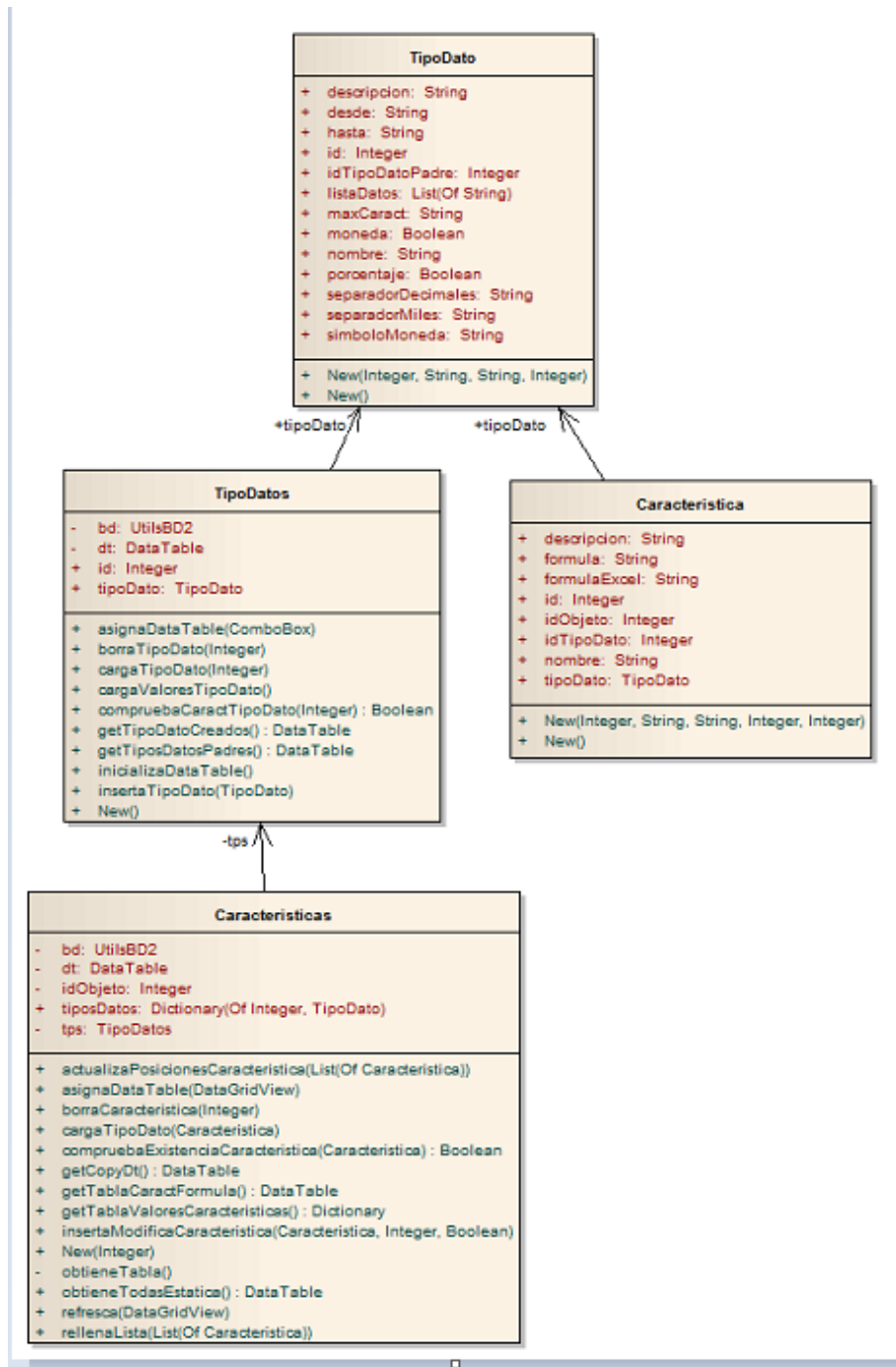


Ilustración 92 - Clases Capa_Objeto_BD 2

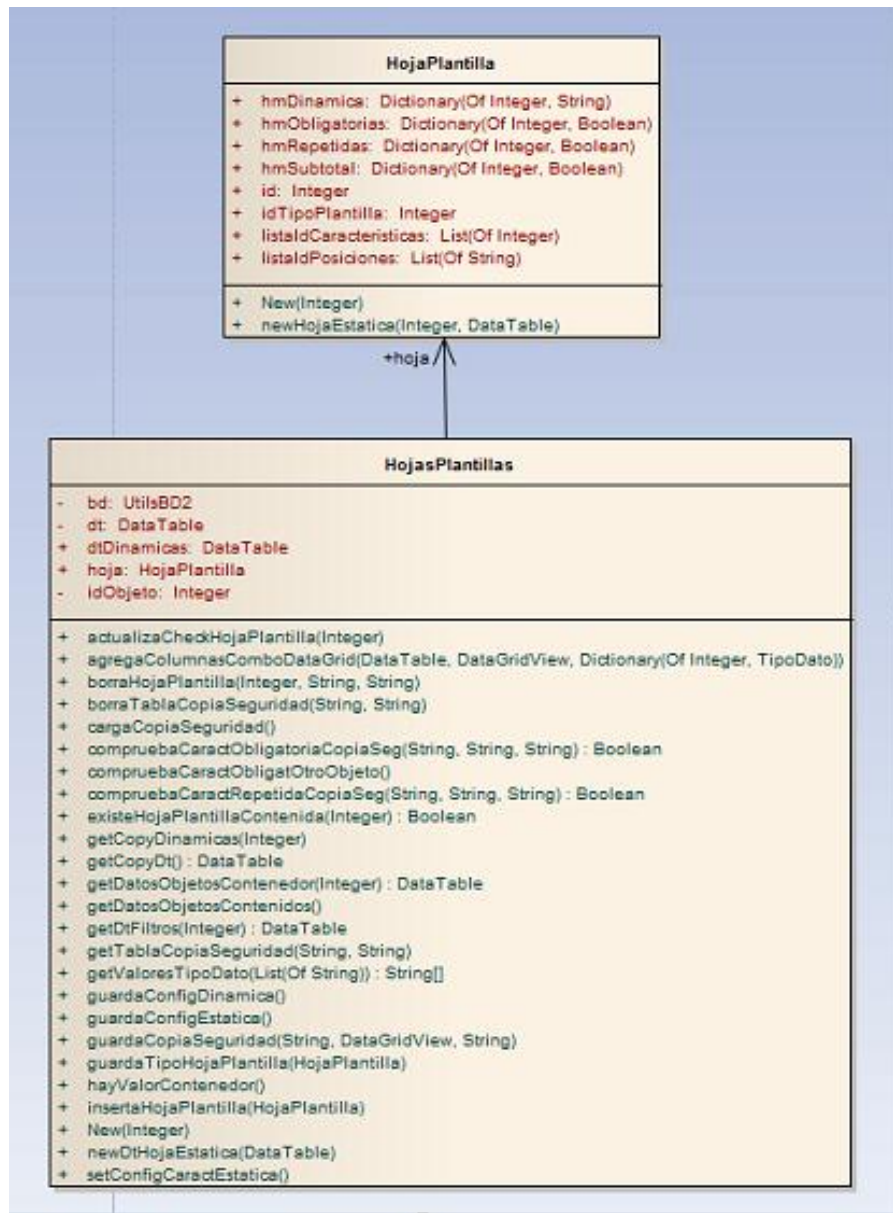


Ilustración 93- Clases Capa_Objeto_BD 3

Capa_Interfaz

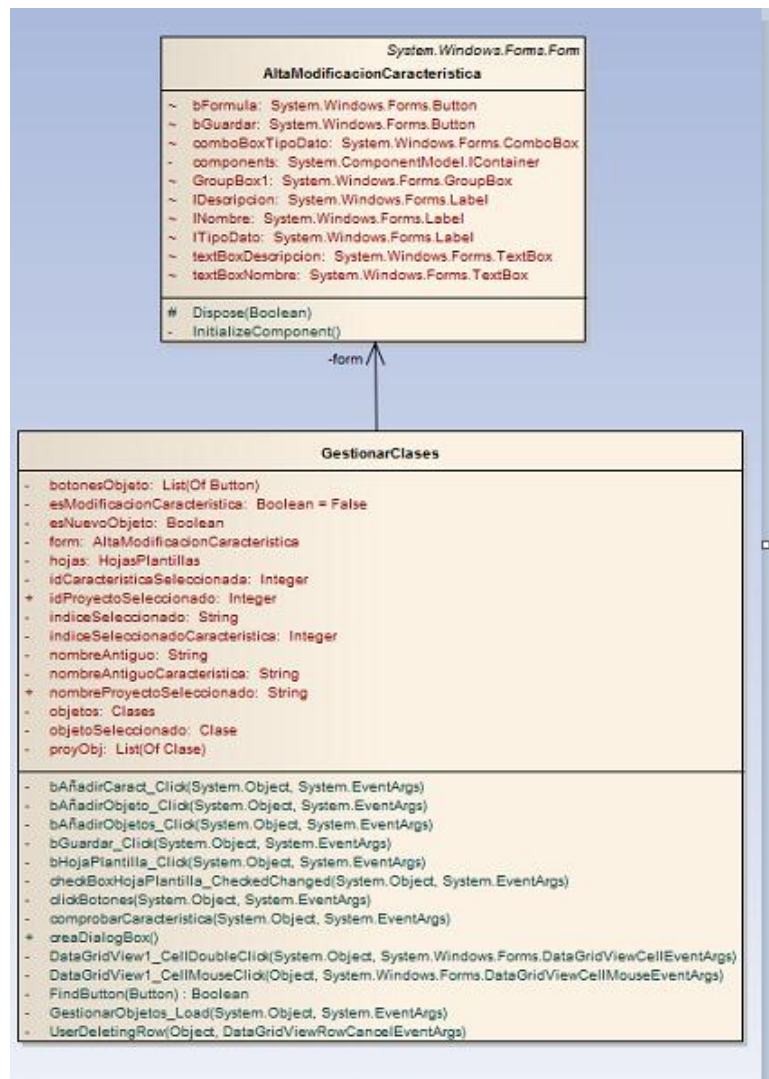


Ilustración 94 – Clases Capa_Interfaz 1

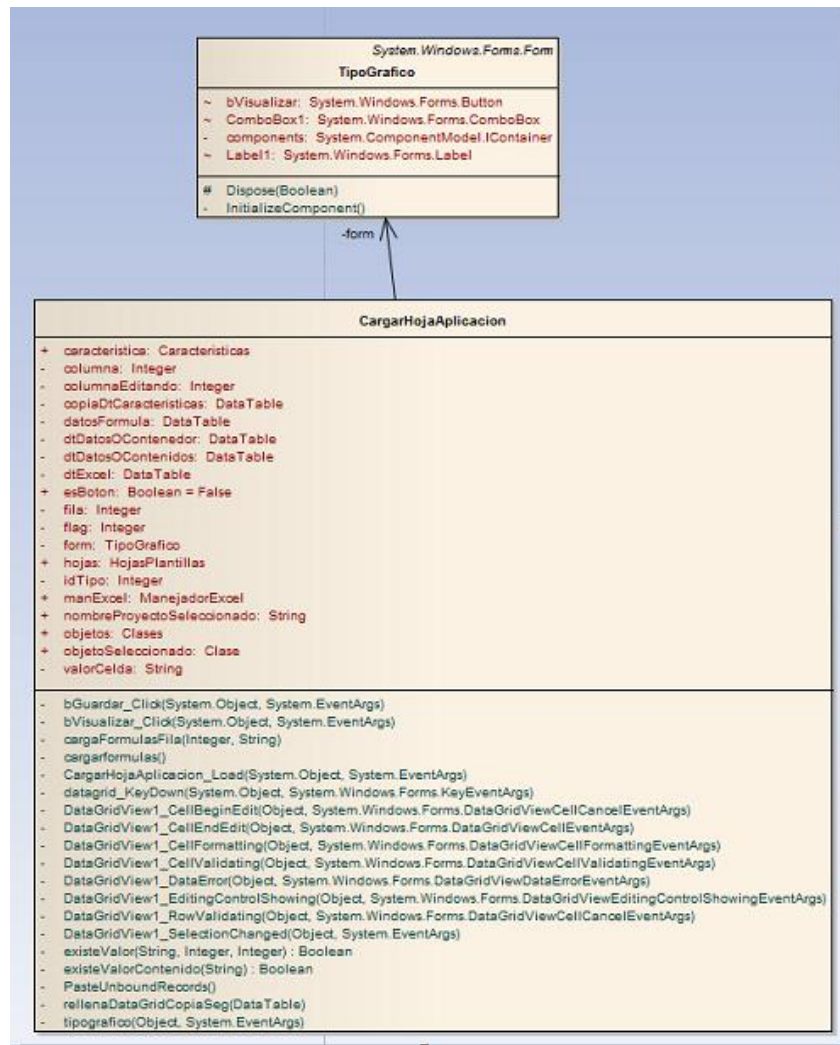


Ilustración 95 - Clases Capa Interfaz 2

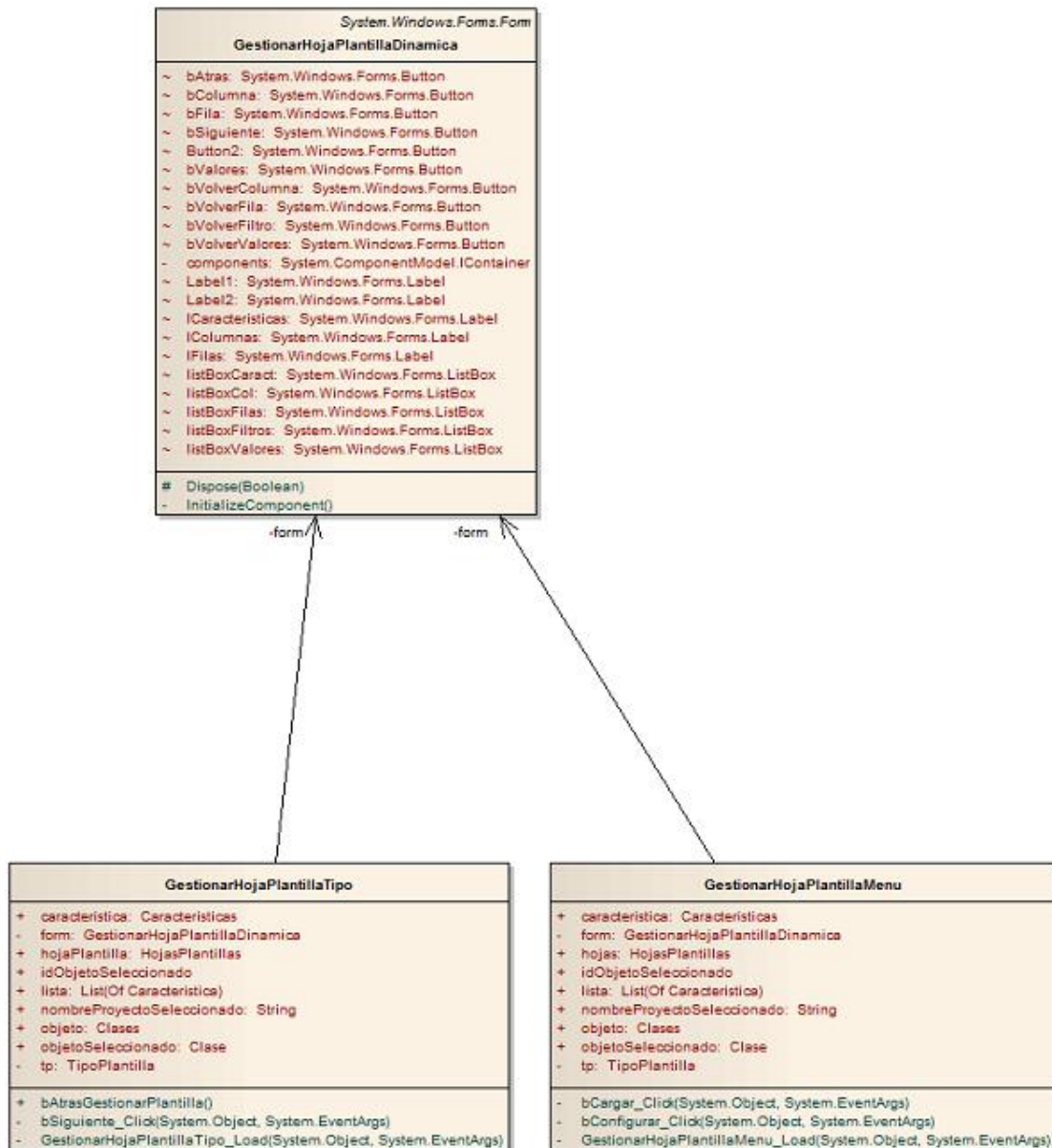


Ilustración 96 - Clases Capa Interfaz 3

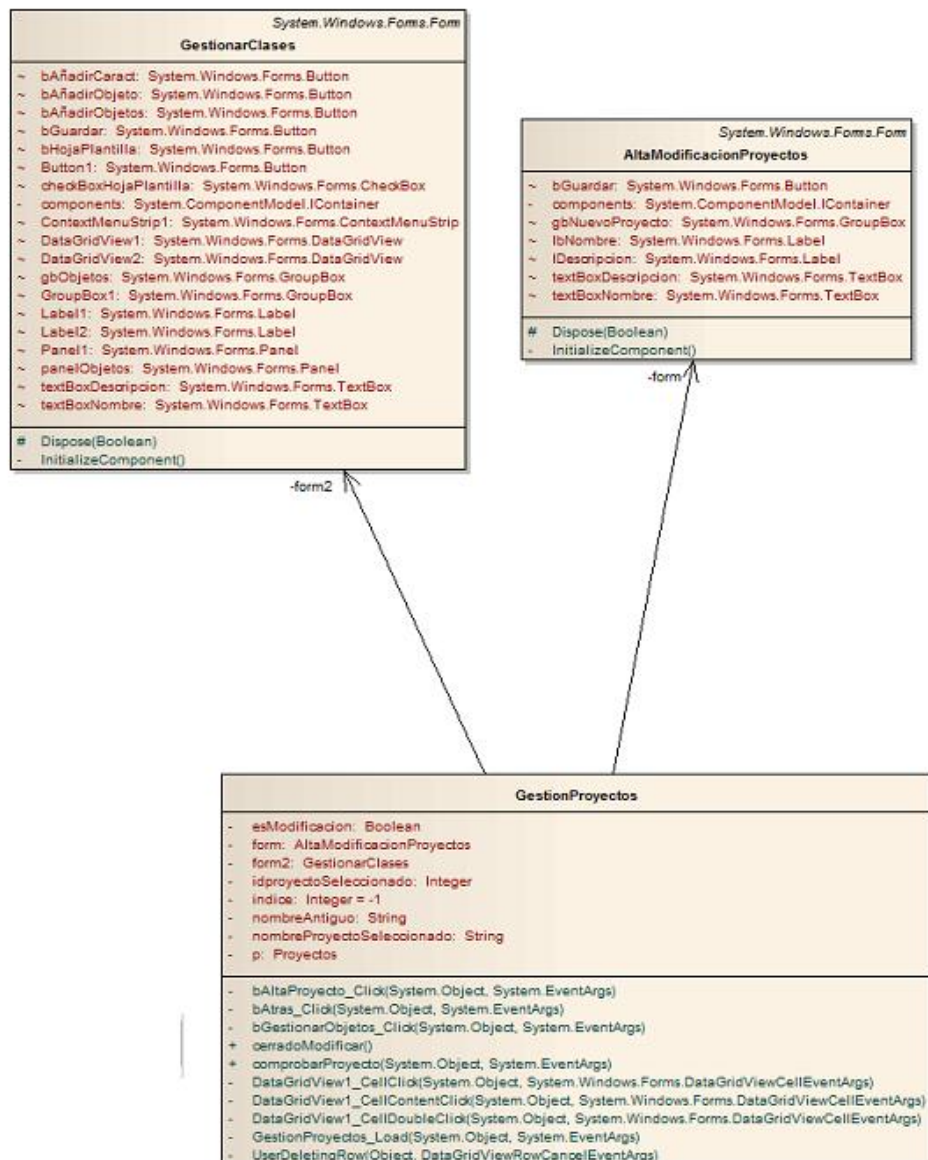


Ilustración 97 - Capa Interfaz 4

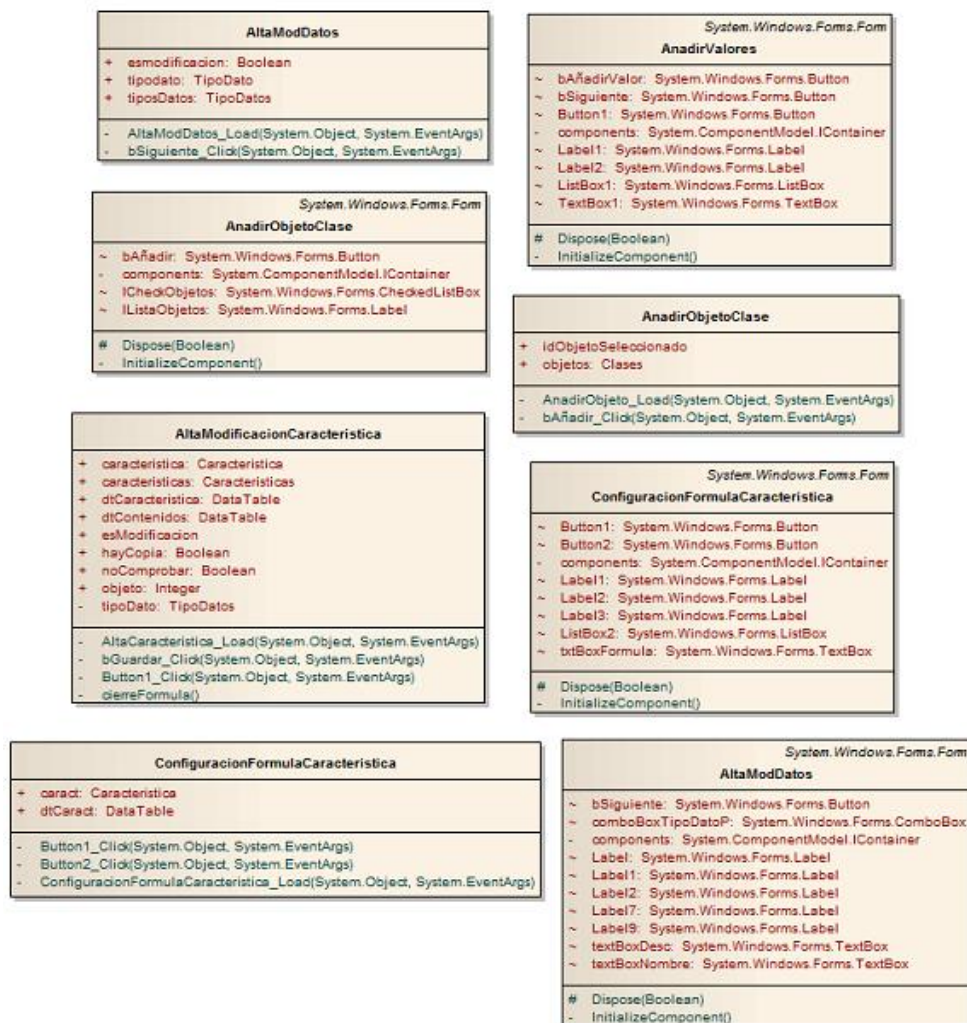


Ilustración 98 - Clases capa interfaz 5

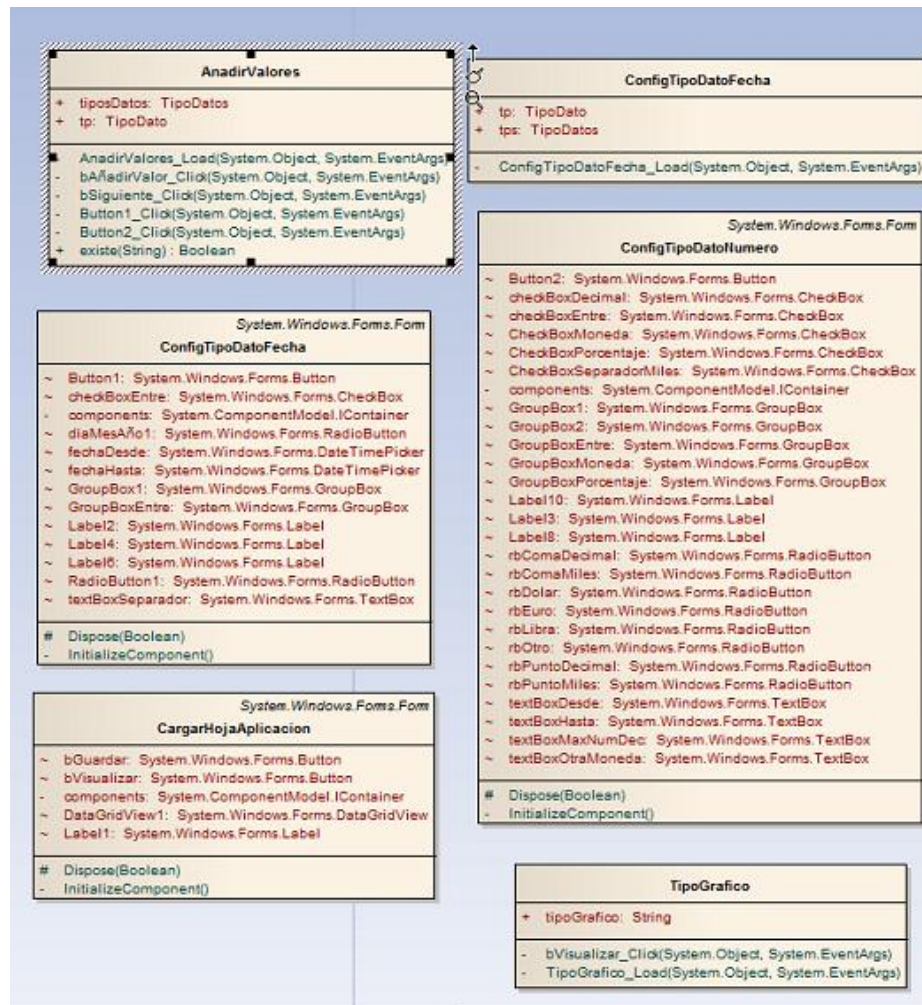


Ilustración 99 - Clases Capa_Interfaz 6

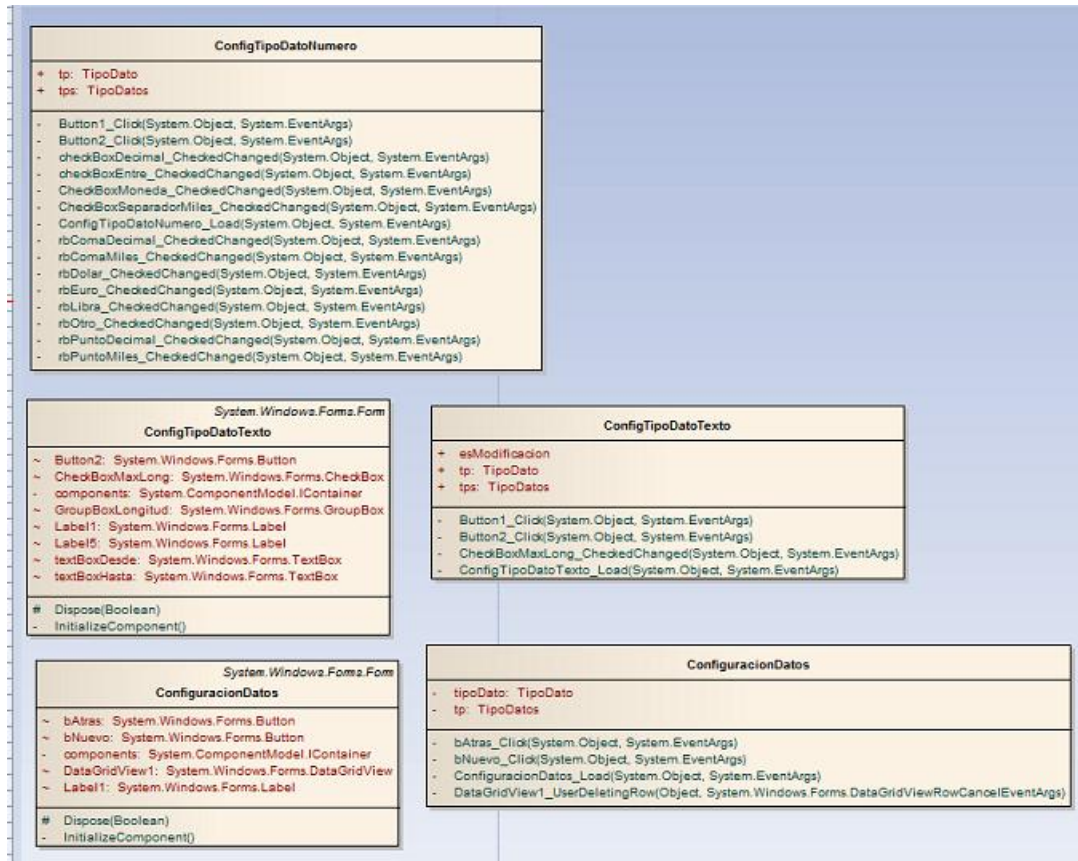


Ilustración 100 - Capa Interfaz 7